EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

A formally verified AKA protocol for vertical handover in heterogeneous environments using Casper/FDR

Mahdi Aiash[1*], Glenford Mapp[1], Aboubaker Lasebae[1], Raphael Phan[2] and Jonathan Loo[1]

Abstract

Next generation networks will comprise different wireless networks including cellular technologies, WLAN and indoor technologies. To support these heterogeneous environments, there is a need to consider a new design of the network infrastructure. Furthermore, this heterogeneous environment implies that future devices will need to roam between different networks using vertical handover techniques. When a mobile user moves into a new foreign network, data confidentiality and mutual authentication between the user and the network are vital issues in this heterogeneous environment. This article deals with these issues by first examining the implication of moving towards an open architecture, and then looking at how current approaches such as the 3GPP, HOKEY and mobile ethernet respond to the new environment while trying to address the security issue. The results indicate that a new authentication and key agreement protocol is required to secure handover in this environment. Casper/FDR, is used in the analysis and development of the protocol. The proposed protocol has been proven to be successful in this heterogeneous environment.

Keywords: authentication and key agreement protocol, secure vertical handover, heterogeneous environments, Casper/FDR

1 Introduction

Future communication systems must allow ubiquitous connectivity where users are always connected from anywhere and at any time. The need for continuous connectivity is being met by the development and deployment of a number of wireless technologies including 3G/HSPDA, WLAN [1] with long term evolution (LTE) [2] and Wimax. However, the widespread deployment of wireless networks will have a significant impact on the evolution of the Internet. However with the wide-scale deployment of wireless networks as end-systems, there will now be significant differences in network characteristics in terms of bandwidth, latency, packet loss and error characteristics. These developments imply that the future Internet will not have a single unified infrastructure. The future Internet comprises a fast core network with slower wireless networks attached around the core. The core network will consist of a super-fast backbone using optical switches and fast access networks which is mainly based on wired technologies such as the multi-protocol label switching (MPLS). Most of the peripheral networks will make use of different wireless technologies. Due to the fact that, the connectivity in the peripheral networks will be based on a wide variety of wireless technologies, provided by different operators, various network operators need to cooperate and coexist in the core network. Furthermore, new providers might choose to join the network and share the spectrum.

Unlike current communication systems such as 2G and 3G, which introduce closed environments where the core network is controlled and owned by sole network operators and thus its security is mainly based on the assumption that, the core network is physically secure, the above discussion highlights the fact that we are moving towards an open, heterogeneous environment where the core network is not controlled by a single operator, so multiple operators will have to cooperate. This tendency will bring about radical changes to the

* Correspondence: M.Aiash@mdx.ac.uk
[1]School of Engineering and Information Systems (EIS), Middlesex University, London NW4 4BT, UK
Full list of author information is available at the end of the article

handover mechanisms. Current mechanisms mainly support the network-controlled handover in which, the decision to implement handover is taken by the network (s) to which the mobile device is currently attached. While, this type of handover works fine in current systems, where the core network is controlled by a sole operator and thus information about the topology of different networks is available, this type of handover is not suitable for heterogeneous environments, since multiple operators coexist in the core network. This highlights the need for the client-based handover in which the client is the deciding entity rather than the network. In this type of handover, the mobile device will be responsible for initiating the handover, acquiring and releasing the resources in the new and old network respectively. However, this situation brings about new security threats in term of authenticating the mobile device to access the new network in case of handover and maintaining data confidentiality as well as controlling the allocation of network resources in case of handover by making sure that this process is accomplished by authorized parties. While the latter issue was addressed by the research in [3], the first has been investigated by different research efforts such as [4-9].

These efforts have considered the openness and dynamic nature of the future networks while designing their security mechanisms. However, some solutions such as the AKA protocols for the 3GPP-WLAN and 3GPP-WiMAx internetworking [8,9] presumed to have the UMTS infrastructure as a backbone of the core network, while different networks such as WLAN and WiMax could be attached to it. Obviously, this solution does not go along with the open architecture of future networks. Other studies such as the AKA protocol of the HOKEY WG [4] proposed to use a common platform such as the extensible authentication protocol (EAP) [10] to hide the differences between the access networks. In contrast, the solution proposed by the mobile ethernet group [7,11] assumes a generic network structure, which is very close to the afore-mentioned open architecture. Therefore, this article will consider the mobile ethernet's vertical handover AKA protocol as a model to investigate the security threats in the open architecture. The protocol will be analyzed and verified using formal methods approach. The results discovered some security breaches in the deployment of the mobile ethernet's AKA protocol, which highlight the need for a new protocol.

Modeling and analysis of security protocols with communication sequential process (CSP) [12] and failure-divergence refinement (FDR) [13] have been proven to be effective in discovering attacks in many protocols such as [14-16]. However, describing protocols in CSP is a quite exhaustive and time-consuming process. Therefore, a new compiler has been introduced in [17]. The compiler is known as Casper, it accepts an abstract description of the protocol and translates it into CSP. In order to verify the security properties of the protocol, the FDR is used to model and analyze the CSP output.

This study adds the following contributions: First, it analyzes some of AKA protocols for handover. Second, it uses the Casper/FDR to formally verify and analyze the handover AKA protocol of the mobile ethernet, the verification discovered authentication attack. Thirdly, to address the discovered drawbacks of the protocols in the literature, a new AKA protocol for secure vertical handover in heterogeneous environments is introduced. A detailed refinement of the protocol is presented with a formal versification of each of the refinement stages using Casper/FDR. We also describe all the attacks found in each stage of the refinement process. The last version of the protocol, is formally verified and proven to achieve many desired security properties.

The rest of this article will be organized as follows: Section 2 views a potential structure of future open networks and describes the IEEE 802.21 research to support vertical handover in heterogeneous environment. Section 3 describes some related research to provide secure vertical handover such as the work of the HOKEY, 3GPP and mobile ethernet groups. Since the mobile ethernet framework considers an open network architecture, Section 4 explains the initial AKA protocol of the mobile ethernet [7] and verifies the protocol using Casper/FDR. The verification results highlights the need for a new AKA protocol. Using a progressive approach, Section 5 explains and formally verifies the refinement stages, which led to the final version of the protocol. The article concludes in Section 6.

2 Network evolution

The next generation networks (NGN) will provide ubiquitous computing via the seamless operation of heterogeneous wireless networks including WLAN, 3G, WiMax, Ultrawideband, etc. Using these networks, users will be continuously connected to the Internet as they move around. Vertical handover which allows mobile nodes to seamlessly switch their connections from one network to another is a key mechanism that must be supported in NGNs. However, in order to effectively support vertical handover there is a need to re-examine the current network structure and define the required changes in the network. These changes need to be reflected in a new networking architecture which attempts to clearly define the functions, their order and the interlocking relationships that are necessary to support heterogeneous networking. Therefore, the following sections describe recent research efforts to define a new structure for future networks to manage the resources in the heterogeneous environment and support the

vertical handover in this environment. They also differentiate between the vertical and horizontal handover, then describe related work to enhance the vertical handover experience either by introducing new vertical handover mechanisms or by addressing the End-To-End QoS and security provision in heterogeneous environments.

3 Overview of future heterogeneous networks

The network infrastructure of NGNs will be owned by different operators. Additionally, new operators could install their network hardware and join the core network. However, interoperability between different operators is a key challenge in this open, heterogeneous environment. To address this issue, the ITU-T recommended deploying a central management entity referred to as the regulatory authority [18], which controls different network operators and service providers. The regulatory authorities are regulatory bodies with the power to influence policies in telecommunication services, they are responsible for creating national policies to encourage the development of telecommunications, also they provide essential powers to regulate license agreements, interconnection arrangements, and monitoring unlawful telecommunication activities.

To enhance the concept of a central management entity, the study of the Y-Comm group [3] and Daidalos II [19] proposed the concept of core end-point (CEP) as an administrative entity to control the different wireless networks in a regional area, as shown in Figure 1.

A detailed view of the CEP's structure along with the attached networks is shown in Figure 2. The figure shows a hierarchical architecture, where the bottom level is represented by several access points (APs) and access routers (ARs) that communicate with the wireless interfaces in the mobile terminals. The middle level comprises a number of technology-specific domains, where each domain represents a certain network operator and technology such as 2G, 3G, and Wi-Fi. For these domains to interoperate, the CEP, which is residing at the top level acts as a central administrative domain to control the inter-domain functions and provide overall management.

Although the structure in Figure 2 is for future network, it can also be used alongside the architecture of current systems; for instance, the technologies-specific domains in the mid-level correspond to the circuit switching and packet switching core networks in the GSM and GPRS or UMTS. The major difference is that the proposed structure is an open architecture, where different technologies and operators could join the network. However, to control this open architecture, the CEP in the top-level has been proposed to manage the resources in all various domains.

In order to deal with the QoS and security tasks in this architecture, a number of operational entities have been proposed as follows:

- *The central A3C server (CA3C):* This is the central authentication, authorization; accounting and cost (A3C) server in the CEP. The CA3C holds the service level of agreements (SLAs) along with the network level of agreements (NLAs), which describe the clients' terms for using the service and accessing networks, respectively.
- *The central QoS broker (CQoSB):* is responsible for negotiating QoS in case of cross-CEP handover.
- *The domain A3C server (DA3C):* The DA3C is responsible for handling users' service aspects. Initially, it extracts users' profile information from the CA3C and uses this information for authorizing the users' requests to access services.
- *The domain QoS broker (DQoSB):* manages the resources of the attached peripheral networks with respect user preferences and network availability, it also makes a per-flow admission control decision.
- *The access router (AR):* This is the link between the domain and the peripheral networks; it enforces the admission control decision, taken by the DQoSB. Since the AR acts as a relay between the mobile terminal (MT) in the peripheral network and the DA3C, using security terminology, the AR will be referred to as the authenticator (Auth).

These entities cooperate to provide security and QoS-related tasks. However, since there is a need for QoS provision in different situations, three QoS-Signalling models have been proposed in [3]:

- *The registration model:* describes the procedure followed when the MT first attaches to the peripheral network. This model basically involves authenticating the MT to use the network, then enforcing the access control policies based on the MT's SLA. This article investigates different AKA protocols and proposes a novel one to be integrated with the registration model.
- *The connection initiation model:* deals with the case when the MT starts a connection to a server SP. It involves authorizing the connection request in both the source and the destination networks and making sure that it complies with the pre- agreed on QoS. Once this is achieved, layer two resources in both networks are prepared to accommodate the connection.
- *The handover model:* This step explains the QoS provision in the case of inter and intra administrative domain handover. This step deploys the authentication

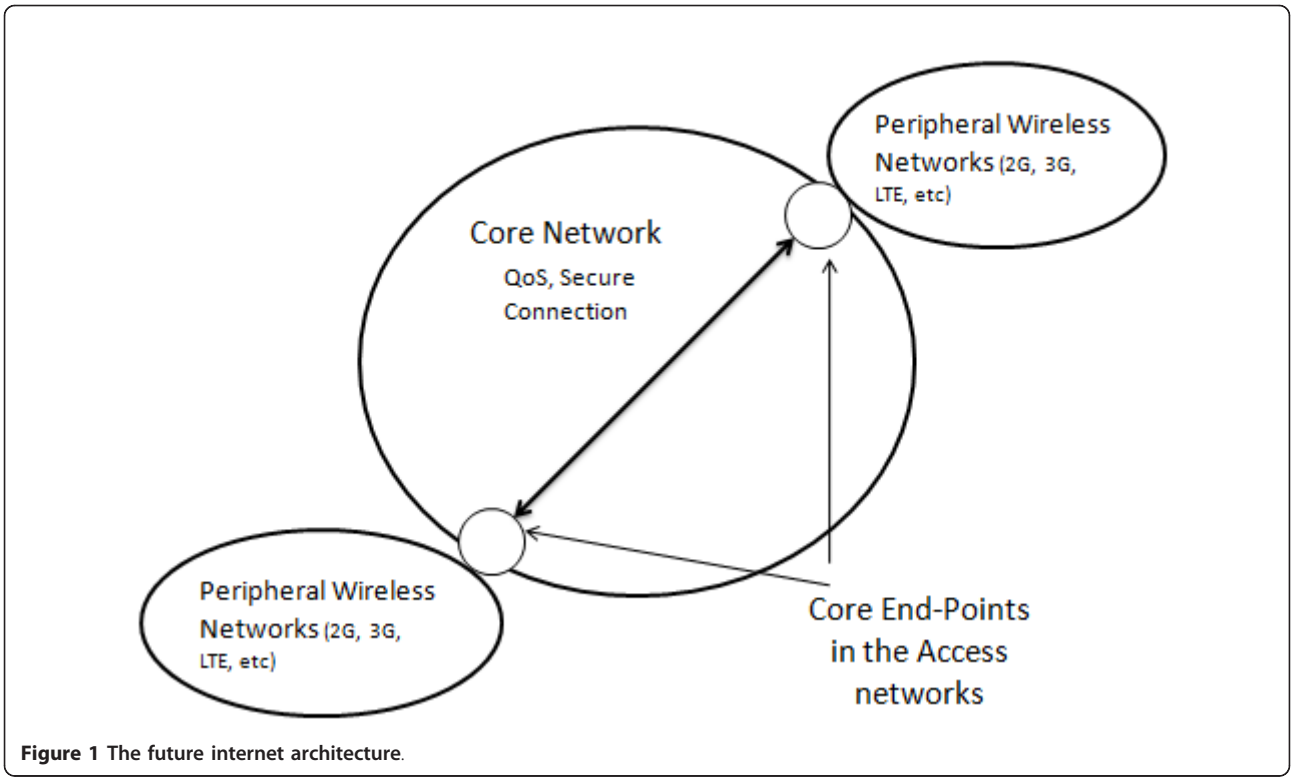

Figure 1 The future internet architecture.

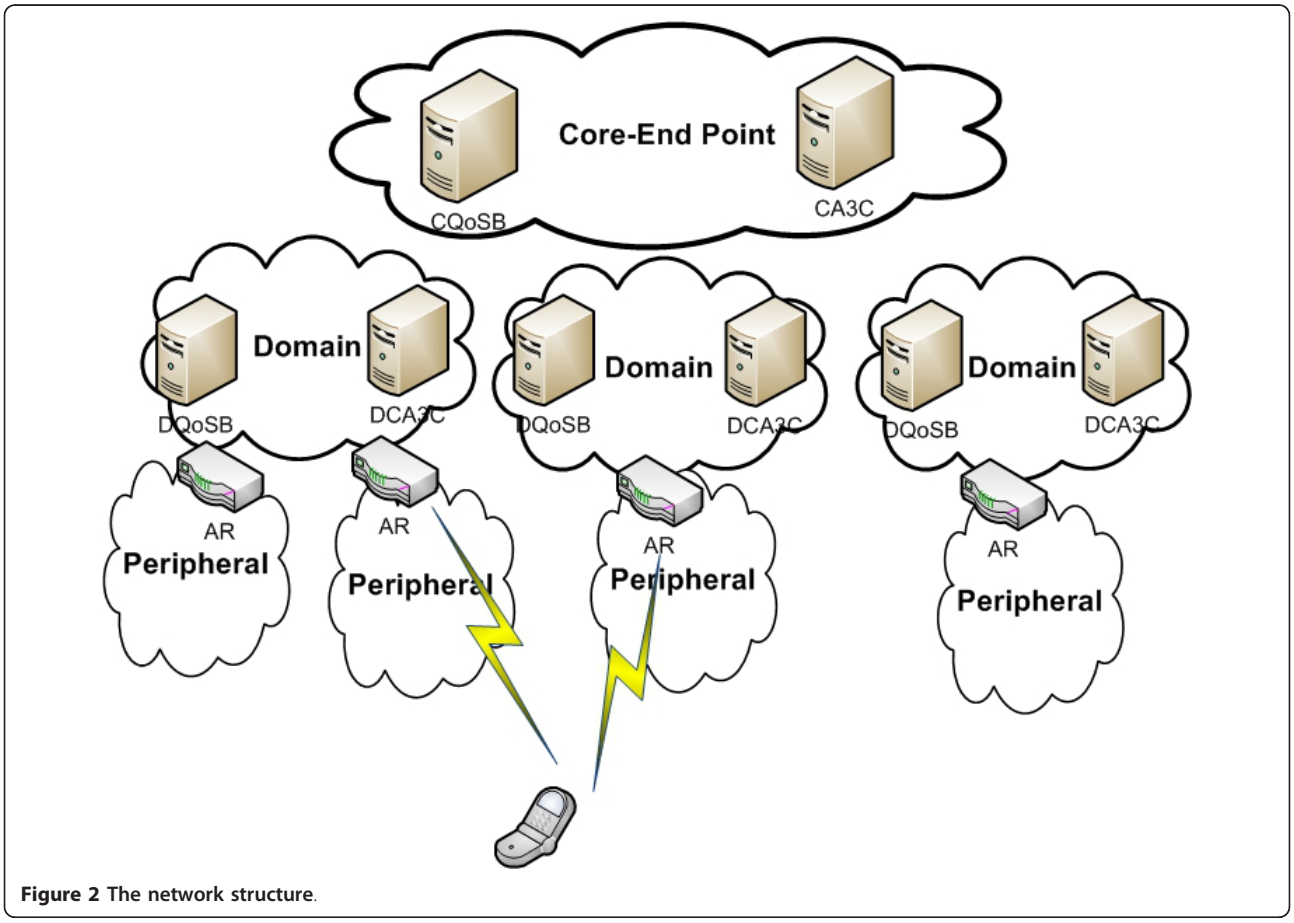

Figure 2 The network structure.

and key agreement (AKA) protocol to achieve pre-authentication and launching the security materials in the target network also in this step, the QoS- context is transferred and used by the access control mechanism in the new network.

More details about these models are found in [3].

3.1 Vertical handover vs horizontal handover

In handover, mobile nodes change the point of attachment from one network AP to another. However, if the mobile node moves within a single technology network, this is known as horizontal handover. So a mobile node in GSM network performs a horizontal handover when it moves from one GSM cell/access point to another. Vertical handover takes place when the mobile node roams between different access technologies switching from GSM to 3G or Wi-Fi for instance. Hence, in heterogeneous environment, where there are many wireless networks operating in the same area, vertical handover will become commonplace. Thus the security threats such as the authentication of mobile devices as well as access to network resources need to be address in order to provide secure vertical handover.

3.2 Vertical handover mechanisms in heterogeneous environments

The IEEE 802.21 working group has developed standards to enable handover and interoperability between heterogeneous network types including both 802 and non 802 networks. As stated in [6], The purpose of IEEE 802.21 is to improve users' experience by providing media independent handover (MIH) functionality that facilitates both mobile-initiated and network-initiated handover.

To optimize handover in heterogeneous environments, the IEEE 802.21 proposes an intelligent and generic interface that operate between the data link (L2) and Network layers of the protocol stack. This interface holds all the required functions to support MIH and thus is referred to as media independent handover function (MIHF).

In the world of the IEEE 802.21, the MIHF should be available in the MT and the network entities. The MIHF encompasses three types of services:

- *MIH event services (MIES)* detect changes in link layer, report them to the upper layers [20]. These events might be used as indicators for a potential handover.
- *Media independent command service (MICS)* provides a set of commands that enables the upper layers (policy or mobility management layers) to control the status of the link such as switching it on or off. Additionally, some of the MICS commands enable the upper layer to ask the link layer about its status before making the handover decision, this is very crucial to support proactive, mobile-initiated handover.
- *Media independent information service (MIIS)* provides information such as topology, location and link layer parameters (data rate, throughput, etc.) about different networks in the vicinity. This information, if provided beforehand, will aid the mobility management protocol on the handover' decision.

The IEEE 802.21 standard provides functions and libraries to support vertical handover in heterogeneous networks. Also, its proposed vertical handover system might be considered as a reference model for other models in any future framework such as the ambient networks [21] and Y-Comm [22].

4 Secure vertical handover in heterogeneous environment

This section discusses some related study, that have been trying to provide AKA protocols to secure vertical handover mechanisms in future networks.

4.1 The handover key working group (HOKEY WG)

The IETF handover keying working group (HOKEY WG) [4] is currently developing solutions to provide a secure, MIH, also called inter-technology handover. The solutions are applicable to wireless access technologies based on the EAP [10], which is an authentication framework that supports multiple authentication protocols; these are referred to as EAP methods. Based on the EAP terminology, three entities are defined: The EAP peer which is the client asking for authentication using an EAP method, the EAP server is the entity that terminates the EAP authentication method with the peer, the EAP servers are often, but not necessarily, co-located with AAA servers. And finally, the EAP authenticator which is the network AP that supports the authentication functionality and enforces access control based on the authentication result.

When a MT moves between different authenticators, it is desirable to avoid a full EAP authentication to support fast handover. Therefore, the HOKEY group proposed a new method for the EAP known as EAP Re-authentication protocol (ERP) [23].

Initially, the MT performs a full normal EAP authentication with the A3C server in its home network. As a result of this authentication, the EAP's keys namely, master session key (MSK) and extended master session key (EMSK) are derived. For the MT to use the ERP protocol with the AP in the target network, it needs to derive a new re-authentication root key, this key is derived using the EMSK and the domain name of the

target network and hence, is called the domain specific root key (DSRK). Using this key, further domain specific keys such as the domain specific integrity key (DsIK) and domain specific re-authentication MSK (DSrMSKs) are derived, these will be used to secure the connection between the MT and the network. Additionally, the possession of the derived keys achieves authentication between the MT and the network.

In order to get the domain name of the target network, the ERP defines two bootstrapping modes: implicit and explicit. The implicit mode assumes the use of link layer specific announcements, called EAP-Initiate/Reauth-Start packets [23] which advertise the local domain name and are issued by ERP-supported APs. However, if the MT misses the announcement, it needs to sends extra messages to probe for the domain name of the target network.

4.1.1 ERP analysis

The HOKEY's work seems fairly stable particularly in terms of keys hierarchy. However, the solutions for keys distribution are still being discussed. Additionally, the ERP extension suffers from some drawbacks which could be summarized as follows:

- Although the ERP is based on the EAP platform, it introduces new messages such as EAP-Finish/Reauth that includes a DSRK and the new domain name. This implies that, all the network entities such as the APs has to be updated or replaced to support this extra message.
- The ERP presumes that the MT will get the domain name either implicitly when receiving the announcement or explicitly by soliciting for it. The authors believe that this step should be part of the handover procedure rather than a part of the security mechanism. Additionally, it is not clear how the MT would communicate with the EAP Re-authentication server in the target network.
- Although, the security consideration section of the [23] provides some analysis of the protocol features, the protocol lacks formal analysis such as using a formal methods approach.
- Implementing the proposed solution requires the network components to support EAP platform, this assumption might be feasible in heterogeneous environment, where the network infrastructure is owned by multiple operators.

All these drawbacks highlight the fact that, the ERP protocol does not go along with the open architecture of the network as presented in Section 4.3.

4.2 The 3rd generation partnership project (3GPP)

The 3GPP project has introduced two scenarios; the 3GPP-WLAN interworking, which is introduced in Release 6 of 3GPP specifications [8] and 3GPP-WiMAX interworking architectures as examples of heterogeneous environments. Both scenarios presume the presence of 3GPP technology in the core network, while WLAN or WiMax technologies are in the peripheral networks.

In the case of WiMAX to WLAN Vertical Handover, the MT invokes EAP-AKA if the WLAN domain is visited for the first time. Otherwise, fast EAP-AKA re-authentication is executed. In the case of WLAN to WiMAX handover, the MT performs the initial network entry authentication protocol (INEA) which is performed as a part of the privacy and key management protocol version 2 (PKMv2) [24], when visiting the domain for the first time. Otherwise, WiMAX RAP is executed [25].

One issue with this approach is that it is fully dependent on specific wireless technology; the 3GPP core network in this case. Whoever wants to add a new wireless access to an existing network will always need to develop a method that integrates wireless access with the 3GPP core infrastructure.

4.3 The handover AKA protocol of the mobile ethernet

The mobile ethernet consists of a core network and wireless access connects to the core network via a layer two switches, called edge switches. Connectivity in the mobile ethernet is based on MAC addresses and hence various kinds of plug-in wireless communication provided by different operators could coexist.

The mobile ethernet has proposed two AKA protocols: the first is used for the initial authentication; when the mobile device joins the network for the first time. The second AKA protocol is responsible for AKA functions in case of handover. The AKA protocols of the mobile ethernet are not technology-specific and do not require platforms such as the EAP and thus could be deployed by any operator. Also, the network architecture, proposed by the mobile ethernet is very similar to the open architecture in Figure 2. Due to these factors, the handover AKA protocol of the mobile ethernet will be act as model to investigate the potential security threats, it will be analyzed in Section 5 using formal methods approach.

4.4 Verifying security protocols using formal methods and Casper/FDR tool

To verify the protocol, we use a form of formal methods approach based on Casper/FDR tool [17]. The Casper tool accepts an abstract, human-friendly description of the system and compiles it into CSP code, suitable for the FDR [13] checker. CASPER's input file consists of eight headers as explained in Table 1:

4.5 Desired security features for AKA protocols

As stated in [26], it is desired for AKA protocols to meet certain security properties. Therefore, a list of

Table 1 The headers of Casper's input file

Header	Description
# Free variables	Defines the agents, variables and functions in the protocol
# Processes	Represents each agent as a process
# Protocol description	Shows all the messages exchanged between the agents
# Specification	Specifies the security properties to be checked
# Actual variables	Defines the real variables, in the actual system to be checked
# Functions	Defines all the functions used in the protocol
# System	Lists the agents participating in the actual system with their parameters instantiated
# Intruder information	Specifies the intruder's knowledge and capabilities

these properties will be used to analyze both the AKA protocol of [7] and our proposed protocol.

1. *Mutual entity authentication:* This is achieved when each party is assured of the identity of the other party [26].
2. *Mutual key authentication:* This is achieved when each party is assured that no other party aside from a specifically identified second party gains access to a particular secret key [26].
3. *Mutual key confirmation:* This requirement means that each party should be ensured that the other has possession of a particular secret key [26].
4. *Key freshness:* a key is considered fresh if it can be guaranteed to be new and not reused through actions of either an adversary or authorized party [26].
5. *Unknown-key share* **resilience**: In the UKS attack the two parties compute the same session key but have different views of their peers in the key exchange [26]. In other words, in this attack an entity A ends up believing she shares a key with B, although this is the case, B mistakenly believes the key is instead shared with an entity $E \neq A$
6. *Key compromise impersonation resilience:* This property implies that if the Intruder compromised the long-term key of one party, he should not be able to masquerade to the party as a different party [26].

5 Secure vertical handover in mobile ethernet

This section describes and formally analyzes the Vertical Handover AKA protocol proposed by Masahiro et al. [7]. The protocol's participants are as follows:

- *The authentication information server (AIS):* manages the subscriber's information, the AIS corresponds to the core A3C (CA3C) server in Figure 3.
- *The authentication server (AS):* authenticates the subscribers based on information retrieved from the AIS. The AS corresponds to the domain A3C (DA3C) server in Figure 3.
- *The entry points (EPs):* represent one end point for wireless communication and represent APs or ARs.
- *The mobile device (M):* is the MT accessing the network.

Masahiro et al. [7], have assumed that, the devices of the core network are securely installed using mutual authentication and data integrity is maintained in the core network, i.e., between the AIS and the AS or between the different ASs. It is also presumed that, the mobile device has already been authenticated in its current (source) network using the initial AKA protocol described in [7].

5.1 The protocol description

By considering the notation in Table 2, the Vertical handover AKA protocol could be explained as follows:

After running the initial AKA protocol in the source network, the mobile device and the AS would have shared the security context that consists of the UID, MS, AK, and SK. In case of a handover, the security context is transferred, over a presumably secure channel from the old AS to the new AS in the destination network. This means that, the security context is always shared between the mobile device and the network, it also implies that, only the SK is re-established on handover, while the re-establishment of the AK and the authentication process happen after the handover. As stated in [7], the SK transferred during the context transfer continues to be used until the new SK is established.

As shown in Figure 3, since both the mobile device and the AS retain the security context, in the case of handover, mobile device's authentication is based on the previous mutual authentication between the device and the old AS.

At the end of the authentication phase, the M and the AS derive a new handover authentication ID (HOAID), which is used to speed up the handover response. So

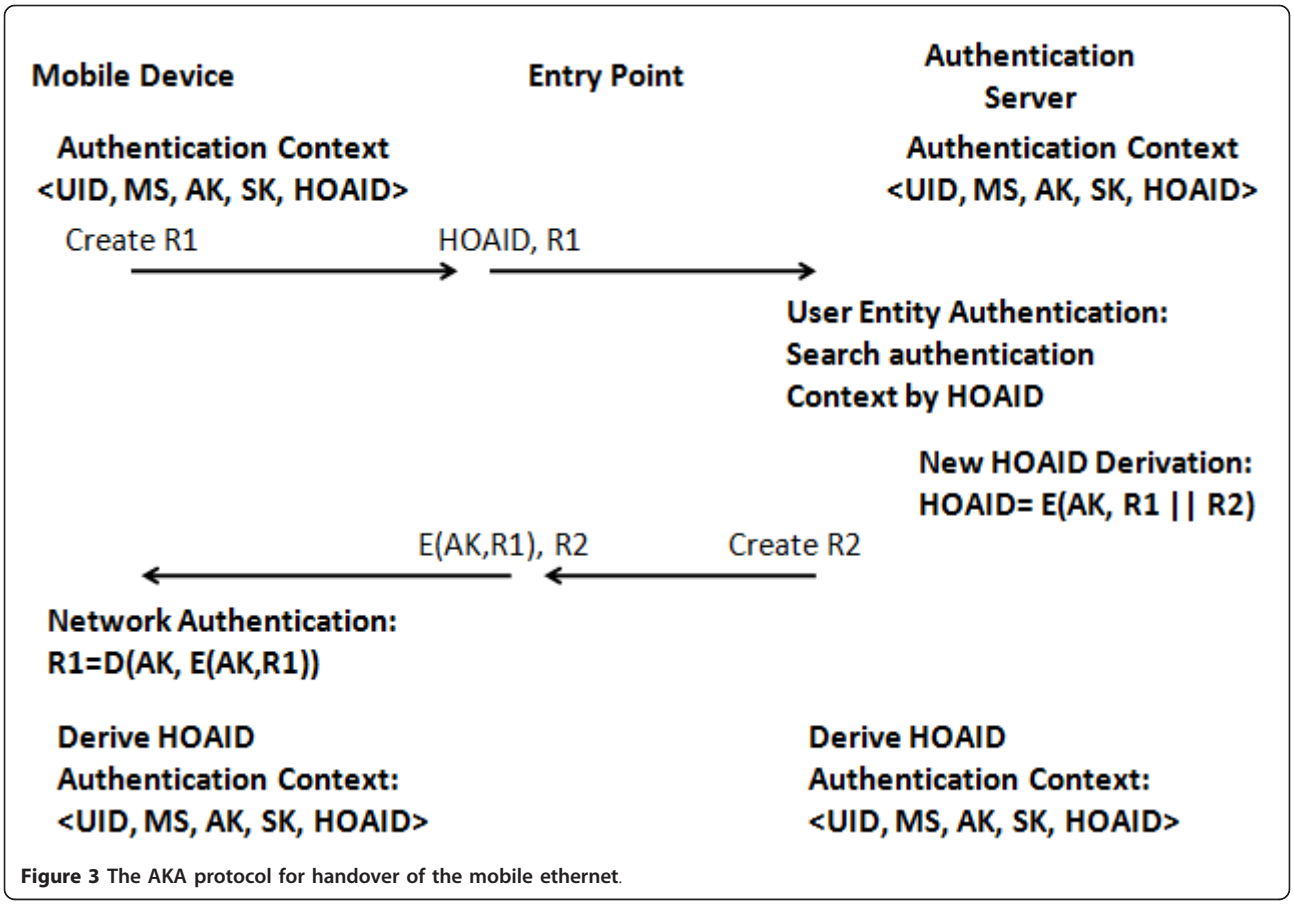

Figure 3 The AKA protocol for handover of the mobile ethernet.

instead of sending the UID, the mobile device will initiates the authentication protocol by sending the HOIAD and the R1 as the first message in Figure 3.

5.2 The formal verification of the mobile ethernet protocol

This section will formally verify the mobile ethernet's AKA protocol for vertical handover using the Casper/FDR tool, then a detailed analysis of the security properties will be introduced. As stated in [7], it is assumed that, the network can trace the movement of the device and determine when handover occurs. However, in order to simulate this using Casper, we introduce the following preliminary messages: the entry point's advertisement messages (Adv), The access request (AccReq) message, which is used by the mobile device to indicate its intention to access the network. The authentication request (AuthReq) message, sent by

Table 2 Notations for the AKA protocol of mobile ethernet

Notation	Description
M	The mobile node
AIS	The authentication information server
AS	The authentication server
R1, R2	Random values
E(K, Msg)	Encrypted Msg by key K
D(K, Msg)	Decrypted Msg by key K
PRF, PRF2	Pseudo-random function
MS	Master secret key MS = PRF(UUK, R1 | R2)
AK	Authentication key AK = PRF(MS, R1 | R2)
SK	Secret key used for encryption SK = PRF2(MS, R1 | R2)
HOAID	Handover authentication ID, an security token for speeding up the authentication in case of handover: HOAID = E(AK, R1 || R2)

the Entry point to trigger the authentication process. None of these messages play a security role; they are only used at the pre-authentication stage, where the EPs advertise their presence.

A Casper input file describing the system in Figure 3 was prepared. The full description is mentioned in Appendix A. for conciseness only the # Processes, the # Specification and the # Intruder Information headings are described here, while the rest are of a less significance in terms of verifying the protocol.

The # Protocol Description section defines the protocol's messages. The notation {m}{k} means that the message (m) is encrypted using the key (k). Also, m%w denotes that the recipient of the message is not supposed to understand the message (m) instead; he should store it in a variable (w) and pass it. In contrast, the notation w%m means that recipient should be able to encrypt the message (m), stored in the variable (w).

The # Processes heading shows that our system comprises four parties: The mobile device (M) is represented by the INITIATOR process, the authenticator process corresponds to EP; the last process namely, the Domain-SERVER represents AS. For each process, the parameters–in the brackets–define the agents' initial knowledge before running protocol.

The security requirements of the system are defined under the # Specification heading. The lines starting with the keyword Secret define the secrecy properties of the protocol. The `Secret (M, SK, [AS, EP])` specifies the SK as a secret between M, EP and AS. The lines starting with Agreement define the protocol's authenticity properties; for instance `Agreement (AS, M, [AK, R1])` specifies that, the AS is correctly authenticated to M using the random number R1 and the AK. The Aliveness assertion checks the availability of the participants, e.g., the first Aliveness check `Aliveness (EP, M)` states that when M completes a run of the protocol, apparently with EP, then EP has previously been running the same protocol. Note that EP may have thought he was running the protocol with someone other than M [17]. A stronger definition of the above Aliveness is specified by the Weak Agreement, for instance `WeakAgreement (EP,M)` assertion could be interpreted as follows: if M has completed a run of the protocol with EP, then EP has previously been running the protocol, apparently with M. Generally, failing to meet the WeakAgreement assertions implies the failure to meet the Aliveness ones.

 # Specification

```
Secret (M, AK, [AS])
Secret (AS, AK, [M])
Secret (M, SK, [AS, EP])
Agreement (AS, M, [AK, R1])
WeakAgreement (M, EP)
WeakAgreement (EP, M)
Aliveness (EP, M)
Aliveness (M, EP)
```

The # Intruder Information heading specifies the intruder identity, knowledge and capability. The first line identifies the intruder as Mallory, the intruder knowledge defines the Intruder's initial knowledge, i.e., we assume the intruder knows the identity of the participants. The last line specifies that the keys of the Domain specific type such as the MS key are crackable. In other words, the crackable keyword tells Casper that, the following keys could be compromised by the intruder at any time of the protocol's run.

After generating the CSP description of the systems using Casper and asking FDR to check the security assertions, two attacks were found. The first discovered attack below is against the `WeakAgreement (M, EP)` and `Aliveness (M, EP)` assertions.

```
0. -> m : ep, as
1a. m -> I_ep : accReq
1b. I_m -> ep : accReq
2. ep -> I_m : authReq
3. I_m -> ep : Garbage
4. ep -> I_as : Garbage, h (Garbage)
5. I_as -> ep : Garbage
6. ep -> I_m : Garbage
```

Figure 4 shows the first discovered attack, which could be described as follows: Initially, the intruder intercepts the connection and replays the messages between EP and M as in messages 1a, 1b, and 2. Pretending to be the mobile device, the intruder composes and fake message with a "Garbage" contents as in message 3. Using this fake message, the protocol continues following the normal sequence and thus, EP completes the run believing it has completed the run with M, while it was with the intruder instead.

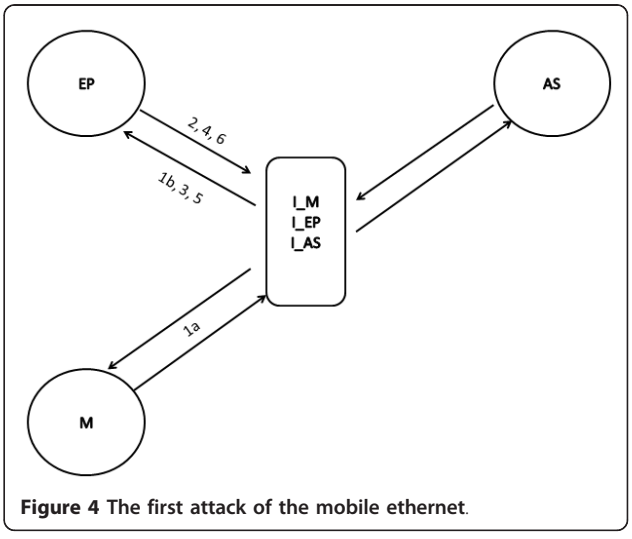

Figure 4 The first attack of the mobile ethernet.

The second attack is against the `WeakAgreement (EP, M)` and `Aliveness (EP, M)` assertions. In this attack (Figure 5), the intruder intercepts and replays the messages between M and EP as in messages 1a, 1b, 2a, 2b, and 3. Once the intruder intercepts message 3, it impersonates EP and completes running the protocol as in messages 4, 5, and 6. Thus, the mobile device will complete running the protocol believing that, it was with EP, while it was with the intruder instead.

```
0. -> m : ep, as
1a. m -> I_ep : accReq
1b. I_m -> ep : accReq
2a. ep -> I_m : authReq
2b. I_ep -> m : authReq
3. m -> I_ep : {m, r1, hoaid1}{sk}
4. I_ep -> as : {m, r1, hoaid1}{sk}, h({m,
r1, hoaid1}{sk})
5. as -> I_ep : {r2, {r1}{ak}}{sk}
6. I_ep -> m : {r2, {r1}{ak}}{sk}
```

5.3 Protocol analysis and security consideration

In this section, we discuss how our formal modeling with Casper allows checking the security requirements described in 4.5.

- *Mutual entity authentication:* In the first discovered attack, the intruder manages to impersonate M to run the protocol with EP. Also, in the second attack, the intruder impersonates EP to run the protocol with the mobile device. These attacks imply that, the protocol does not fulfill this security requirement. These attacks could be ascribed due to the fact that the protocol does not consider verifying the identity of the participants.
- *Mutual key authentication:* the AS is authenticated to M by proving the possession of the random value R1 and the authentication key (AK). We got Casper to check this using the Secret (M, AK, [AS]) and Secret (AS, AK, [M]) assertion checks. Since no attack was found against the key secrecy, this property is met.
- *Mutual key confirmation:* Casper verifies one direction of this requirement by using the decryptable (m, K) which checks if the message (m) is decryptable by the key (K). We performed a similar check after message 6 as shown in the Protocol Description heading to verify that the valid AK is possessed by the AS. If the check fails the protocol aborts. For the mutual authentication, it was presumed in [7] that, the AK along with the security context were transferred from the old AS before the protocol starts, thus there is no need to check this using Casper.
- *Key freshness:* Since the keying materials are transferred from the old AS, this property could be verified by considering the key derivation functions (KDFs) for the $MS = PRF(UUK, R1|R2)$, $AK = PRF(MS, R1|R2)$ and $SK = PRF2(MS, R1|R2)$ in the initial AKA protocol. We could claim that, this property is guaranteed since fresh random values R1, R2 are included in the KDFs of the MS, AK and SK keys.
- *Unknown key share:* The second, discovered attack implies that the UKS was not met. Despite of the fact that, the mobile device (M) and the AS share the AK, the M mistakenly believes that the intruder holds this key as well. Casper/FDR indicates this fact by highlighting an attack against the `WeakAgreement(EP,M)` and `Aliveness(EP,M)` assertions in the # Specifications header.
- *Key compromise impersonation resilience:* this property could be modeled by specifying the long-term keys as crackable and then checking the authenticity assertions. By specifying the MS key to crackable and checking the `Agreement(AS, M, [AK, R1])` assertion, Casper verifies no breach against this authenticity feature.

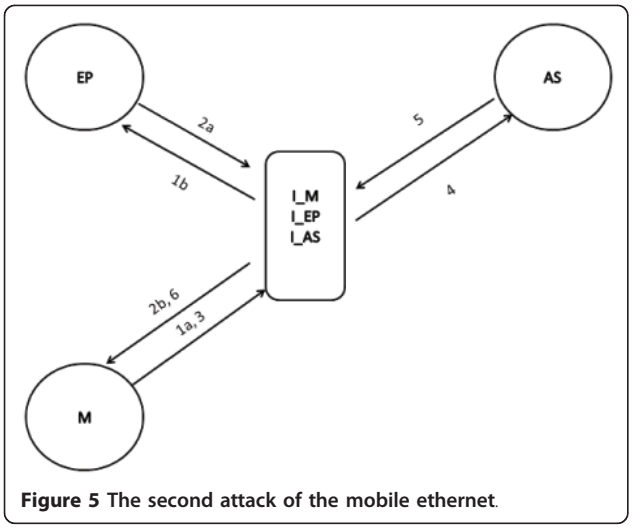

Figure 5 The second attack of the mobile ethernet.

It is obvious that, the mobile ethernet's AKA protocol for vertical handover fulfillled the mutual key authentication, key freshness, and the key compromise impersonation resilience requirements. While it failed in meeting the mutual entity authentication and the unknown key share. Other requirements such as mutual key confirmation could only be achieved if we considered the protocol pre-assumptions of a secure transfer of the security context from the previous AS. This analysis goes along with the verification results of Casper/FDR, where two authenticity attacks were discovered.

This situation highlights the fact that, the assumptions of mutually authenticated entities in the core network

and the integrity of the connection between them, i.e., between the old and new ASs were not efficient, which raises the issue of the need for providing a better security in the core network. In the current systems such as 2/3G, the core network has been assumed to be physically secure, this assumption was valid in this closed, homogeneous environment, where the core network was controlled by a sole operator. However, this assumption does not hold in the case of future networks, where the core network represents an open, multi-operators environments. Additionally, there is a need to deal with identification-related attacks to meet the mutual entity authentication as well as the Unknown Key Share properties.

Furthermore, the process of deriving the keying materials in the Initial AKA protocol of [7] does not define the keys' usability scope. Therefore, there is a need to propose a more stable key hierarchy that specifies the scope of each derived key.

6 The proposed solution

In order to address the previous security threats, this section introduces a new AKA protocol for Vertical Handover in open, heterogeneous environments similar to the one in Figure 2, the new protocol considers the security in the core network at the design stage. However, instead of making assumptions of a secure core network, we need to define the part of the core network to be protected and the type of security mechanism. Therefore, in order to design the proposed protocol, a progressive design approach has been followed; in the initial draft considered in Section 6.3, security was considered in the core network between the CA3C and the DA3Cs, modeling the proposal found secrecy and authenticity attacks, which highlight the main source of threats. The second version discussed in Section 6.4 simulated the case of a secure channel only between the DA3C and the Auth, the discovered attacks in this draft highlight the need to secure different part of the core network. In the final version discussed in Section 6.5, secure channels have been presumed between the DA3C and the Auth as well as between the DA3C and the CA3C. After simulating this case using CSP, Casper failed to find any attacks. This implies that, to address the afore-discovered security threats, the connections between all the entities in the core network have to be protected.

6.1 Defining the security system

The proposed protocol considers the network structure in Section 2. It is crucial to show the actual parties participating in the protocol and thus, how the proposed protocol could be mapped to actual entities in the network.

As shown in Figure 2, the system comprises four entities: the MT performing inter and intra handover, the source and destination authenticators which run on the ARs and presents the MT to the core network; the domain A3C server (DA3C) in the source and destination domains, which are responsible for authenticating and authorizing the MT to use the network, and the central A3C (CA3C) server residing in the CEP.

6.2 The key hierarchy

As explained in Section 5.3, the mobile ethernet AKA protocol does not provide a stable key hierarchy which specifies the keys' usability scope. Therefore, the proposed protocol in this article adopts a clear key hierarchy as shown in Figure 6. Similar to the key hierarchy in GSM and UMTS [27], a top level unique key uk(MT) is stored in the SIM card and is never used for encryption purposes rather it is used for deriving further security keys. The second level key is the domain specific master key (DSMS), as the name implies, this key is unique at the domain level and is derived using an irreversible function F1 as follows: DSMS=F1(uk(MT), seq1, Auth_Domain_Name), where seq1 is a fresh sequence number, the Auth_Domain_Name is the corresponding domain name. Since each domain might have more than one authenticator, the MT could join the domain via any of its Auths, thus, a different secret key (SK) has to be used for each authenticator. One AK is used for mutual authentication between the MT and the network. Similar to F1, two irreversible function F2 and F3 are used to derive AK and SK as follows: AK = F2 (seq1, DSMS), SK = F3(seq1, AuthID, DSMS). Where AuthID is the ID of the Auth and is broadcasted by the Auth in the form of AuthID@DomainName. Defining the KDF used by F1-F3 functions is beyond the scope of this article.

6.3 The initial version of the protocol

The initial version of the protocol, shown in Figure 7, considers the presence a certain trust relationship

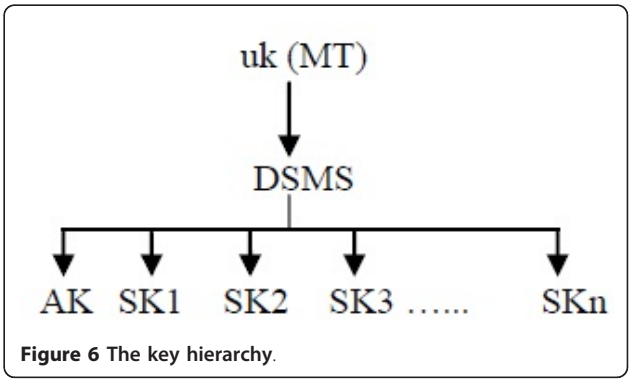

Figure 6 The key hierarchy.

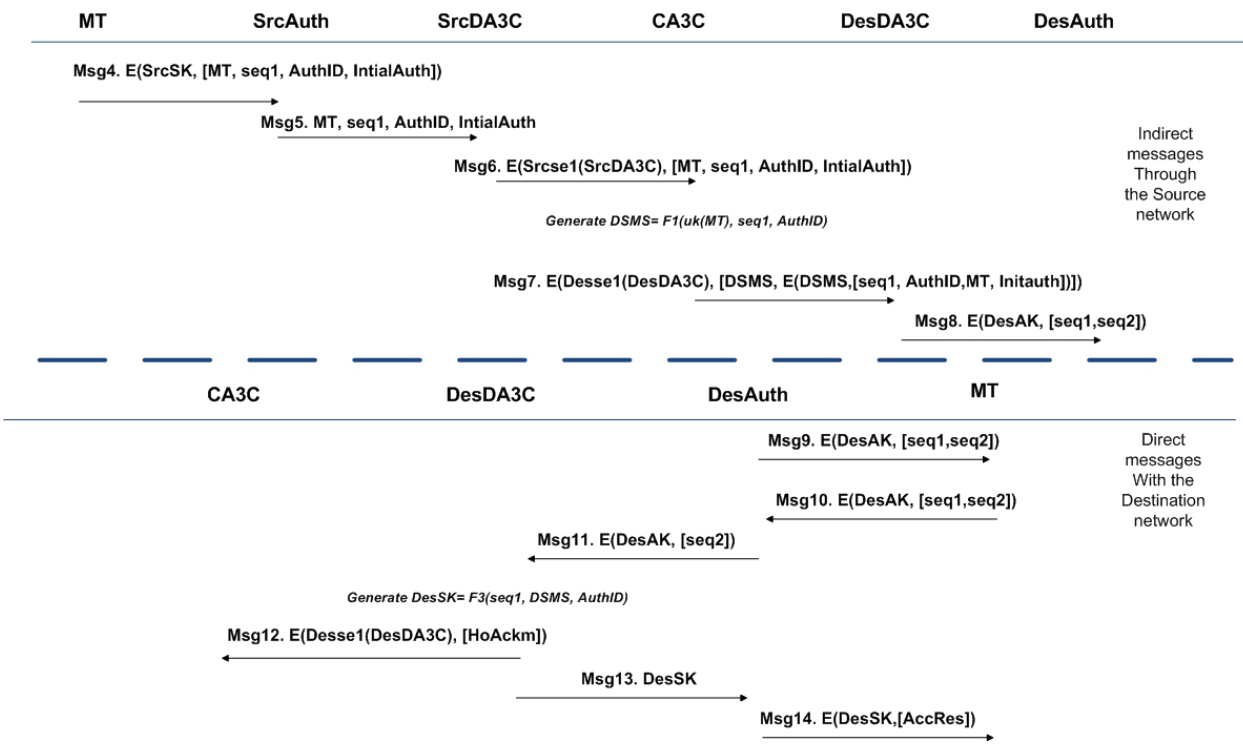

Figure 7 The first version of the proposed AKA protocol.

between the network's entities and thus secure channels have already been established between the CA3C and the DA3Cs. Such secure channels could be guaranteed by using different mechanisms such as IP security (IPSec) [28] or any other virtual private network (VPN) protocols. Alternatively, this could be achieved using out-of-band approach such as agreeing on security materials among the multiple operators. It worth noting that, only security-related messages (starting from Msg 4) are shown in Figure 7. Also as a point of clarification, the protocol's transactions have been split into two groups; indirect transactions which run over the source network (Msg 4-8) and direct transactions with the destination network (Msg 9-14).

To simulate this secure connection between the CA3C and the DA3C using Casper/FDR, a SK se1(DA3C) is presumed to be pre-shared between these entities. Thus, the connections between the CA3C and the DA3C in the source network (SrcDA3C) and the DA3C in the destination network (DesDA3C) are protected using the Srcse1(SrcDA3C) and Desse1(DesDA3C), respectively.

By considering the notations in Table 3, the MT residing in the source network picks the ARs' advertisements (Adv) which contain information about the destination access network such the AuthID and the domain name. The MT uses this information to generate a DSMS.

Phase 1

$Msg1 : DesAuth \rightarrow MT : Adv$

Generate the DSMS= F1(uk(MT), seq1, AuthID)

The protocol starts when the MT sends a joining message Msg 2 to the authenticator in the destination network (DesAuth). The DesAuth responds by sending AuthReq as Msg 3.

Phase 2

$Msg2.MT \rightarrow DesAuth : AccReq$

$Msg3.DesAuth \rightarrow MT : AuthReq$

By using the DSMS, the MT derives a new AK in the destination network (DesAK) and composes Msg 4, this message consists of a fresh sequence number seq1 used as a challenge, authentication ID (AuthID); the MT identity, and an unset Initauth flag (InitAuth = 0). Since the MT has already been authenticated in the source network, the connection with the SrcAuth will be encrypted using the source secret key (SrcSK). The SrcAuth passes this message to the SrcDA3C and from there to the CA3C as Msgs 5 and 6. Using the included mobile ID, the CA3C looks up the corresponding uk (MT) and uses it to generate a fresh DSMS.

Phase 3

Generate the DesAK = F2(seq1, DSMS)

$Msg4.MT \rightarrow SrcAuth : \{MT, seq1, AuthID, Initauth\}_{SrcSK}$

$Msg5.SrcAuth \rightarrow SrcDA3C : seq1, AuthID, Initauth$

$Msg6.SrcDA3C \rightarrow CA3C : \{MT, seq1, AuthID,$

Table 3 Notation

Notation	Description
MT	The mobile terminal
SrcAuth	Is the access router in the source peripheral network
DesAuth	Is the access router in the destination peripheral network
AuthID	The authenticator unique ID has the format AuthID@domainname
SrcDA3C	The DA3C server in the source domain
DesDA3C	The DA3C server in the destination domain
CA3C	Core-endpoint entity, which has QoS and security related responsibilities
Srcse1(SrcDA3C)	Pre-shared secret key between the CA3C and the SrcDA3C
Desse1 (DesDA3C)	Pre-shared secret key between the CA3C and the DesDA3C
Srcse2(SrcAuth)	Pre-shared secret key between the SrcDA3C and the authenticator (SrcAuth)
Desse2 (DesAuth)	Pre-shared secret key between the DesDA3C and the authenticator (DesAuth)
uk(MT)	Unique secret key shared between the CA3C and the MT
DSMS	Domain specific- master key DSMS= F1 (uk(MT), seq1, auth-domain name)
SrcAK, DesAK	The authentication key in the source and destination domains
SrcSK, DesSK	The secret key in the source and destination networks, respectively. These are used to encrypt the connections between the MT and the authenticators
F1, F2, F3	Irreversible key derivation functions
InitAuth flag	A flag set only in the initial authentication. In case of handover, this flag will not be set
HoAckm	Joining/handover acknowledgement message used by the DA3C server to inform the CA3C in the CEP about a successful authentication
seq1, seq2	Sequence numbers
$\{m\}_K$	Encrypting the message (m) using the key (K)

$Initauth\}_{Srcse1(SrcDA3C)}$
Generate the DSMS= F1(uk(MT), seq1, AuthID)

The DSMS key is included in Msg 7, which is sent over the secure channel using the pre-shared Desse1 (DesDA3C) key. Using the information in this message, the DesDA3C generates the authentication key (DesAK) and returns the previously sent sequence Seq1 and a new sequence Seq2 all the way to the MT as Msgs 8 and 9. These messages are encrypted using the derived DesAK. Since the MT has the required information to derive all the keys (DSMS, DesSK, DesAK), the MT verifies the contents of Msg 9 and derives the secret key DesSK.

Phase 4

$Msg7.CA3C \rightarrow DesDA3C : \{DSMS, \{seq1, AuthID, MT, Initauth\}_{DSMS}\}_{Desse1(DesDA3C)}$
Generate the DesAK = F2(seq1, DSMS)
$Msg8.DesDA3C \rightarrow DesAuth : \{seq1, seq2\}_{DesAK}$
$Msg9.DesAuth \rightarrow MT : \{seq1, seq2\}_{DesAK}$
Verify the message contents, then derive the DesSK:= F3(seq1, DSMS, AuthID)

The MT returns Seq2 all the way to the DesDA3C as Msgs 10 and 11. The DesDA3C verifies the contents of Msg 11 and derives the secret key DesSK.

Phase 5

$Msg10.MT \rightarrow DesAuth : \{seq1, seq2\}_{DesAK}$
$Msg11.DesAuth \rightarrow DesDA3C : \{seq2\}_{DesAK}$
Verify the message contents, then derive the DesSK:= F3(seq1, DSMS, AuthID)

Upon verifying the Msg 11, the DesDA3C authenticates the MT and acknowledges this to the CA3C, and then generates the Secret Key (DesSK) and passes it to the DesAuth in Msgs 12 and 13. Using the DesSK, the DesAuth sends an encrypted access response message to the MT as Msg 14.

Phase 6

$Msg12.DesDA3C \rightarrow CA3C : \{HoAckm\}_{Desse1(DesDA3C)}$
$Msg13.DesDA3C \rightarrow DesAuth : DesSK$
$Msg14.DesAuth \rightarrow MT : \{AccRes\}_{DesSK}$

6.3.1 Formal verification

A Casper description of the protocol was prepared. However, since this is an initial version of the protocol, only the #Specifications heading is mentioned here. A complete description of the final and completely refined version of the protocol will be included in the Appendix B.

 # Specification

```
Secret (MT, DesAK, [DesDA3C])
Secret (DesAuth, DesSK, [MT, DesDA3C])
Agreement (MT, DesDA3C, [seq2])
Agreement (DesDA3C, MT, [seq1, DesAK])
WeakAgreement (MT, DesAuth)
WeakAgreement (DesAuth, MT)
WeakAgreement (DesAuth, DesDA3C)
```

```
WeakAgreement (DesDA3C, DesAuth)
Aliveness (MT, DesAuth)
Aliveness (DesAuth,MT)
```

After modeling the protocol using Casper and checking the corresponding CSP code using FDR checker, the following attacks were discovered:

The first attack is against the `Secret(DesAuth, DesSK, [MT, DesDA3C])` assertion, where the Intruder launches a replay attack and eventually manages to get the secret key (SK). The message sequence involved in the attack is given below.

```
0. -> mt : srcAuth, desAuth, srcDA3C
1a. desAuth -> I_mt : adv, desDA3C
1b. I_desAuth -> mt : adv, desDA3C
2a. mt -> I_desAuth : accReq
2b. I_mt -> desAuth : accReq
3a. desAuth -> I_mt : authReq
3b. I_desAuth -> mt : authReq
4a. mt -> I_srcAuth : {SEQ1, authID, mt,
initauth}{srcSK}
4b. I_mt -> srcAuth : {SEQ1, authID, mt,
initauth}{srcSK}
5a. srcAuth -> I_srcDA3C : SEQ1, authID,
mt,
initauth
5b. I_srcAuth -> srcDA3C : SEQ1, authID,
mt,
initauth
6a. srcDA3C -> I_ca3c : {SEQ1, authID, mt,
initauth}{Srcse1(srcDA3C)}
6b. I_srcDA3C -> ca3c : {SEQ1, authID, mt,
initauth}{Srcse1(srcDA3C)}
7a. ca3c -> I_desDA3C : {DSMS, {SEQ1,
authID, mt, initauth}{DSMS}}{Desse1
(desDA3C)}
7b. I_ca3c -> desDA3C : {DSMS, {SEQ1,
authID, mt, initauth}{DSMS}}{Desse1
(desDA3C)}
8a. desDA3C -> I_desAuth : {SEQ2, SEQ1}
{DesAK}
9a. I_desAuth -> mt : {SEQ2, SEQ1}{DesAK}
10a. mt -> I_desAuth : {SEQ2}{DesAK}
8b. I_desDA3C -> desAuth : {SEQ2, SEQ1}
{DesAK}
9b. desAuth -> I_mt : {SEQ2, SEQ1}{DesAK}
10b. I_mt -> desAuth : {SEQ2}{DesAK}
11a. I_desAuth -> desDA3C : {SEQ2}{DesAK}
11b. desAuth -> I_desDA3C : {SEQ2}{DesAK}
12. desDA3C -> I_ca3c : {hoAckm}{Desse1
(desDA3C)}
13a. desDA3C -> I_desAuth : DesSK
13b. I_desDA3C -> desAuth : DesSK
14. desAuth -> I_mt : {accRes}{DesSK}
The intruder knows DesSK
```

The attack shown in Figure 8 could be explained as follows:

- The intruder intercepts the messages between the MT and the DesAuth as in messages 1a, 1b, 2a, 2b, 3a, and 3b. The MT responds by starting the protocol in the normal sequence and sends message 4a.
- The intruder passively intercepts and replays the messages in the destination domain as messages (4a, 4b, 5a, 5b, 6a, 6b, 7a, and 7b). Upon intercepting message 8a, the intruder starts a new session and thus the intruder plays two roles as follows:
  1. Acting as the DesAuth, the intruder replays message 8a towards the MT as message 9a. The MT, mistakenly believing that it is dealing with the DesAuth, replies by sending message 10a towards the DesAuth. However, this message will be blocked by the intruder and replayed later as message 11a. Upon verifying this message, the DesDA3C mistakenly authenticates the intruder, acknowledges this to the CA3C and sends the newly derived DesSK as in message 12 and 13a respectively. These messages will be blocked by the intruder.
  2. The reason behind the second run is to make the DesAuth believes that it is part of the protocol. Therefore, acting as the DesDA3C, the intruder replays message 8a as message 8b towards the DesAuth, which responds by sending message 9b towards the MT. This message is blocked by the intruder so the MT will not have duplicate message. As a response to message 9b, the user, acting as the MT, replays message 10a (from session 1) as message 10b towards the DesAuth. Mistakenly believing it is running the protocol, the DesAuth passes message 10b towards the DesDA3C as message 11b, which is blocked by the intruder so the DesDA3C will not have a duplicate messages and thus discover the attack.
- To complete the protocol and close it stealthy, the intruder passes the DesSK in message 13b to the DesAuth, which composes the AccReq towards the MT in message 14. This message will be blocked by the intruder.

The second attack is against the `WeakAgreement (Des- Auth, DesDA3C)` assertion, where the Intruder launches a replay attack and successfully impersonates the DesAuth. The message sequence involved in the attack is given below.

```
0. -> mt : srcAuth, desAuth, srcDA3C
1a. desAuth -> I_mt : adv, desDA3C
1b. I_desAuth -> mt : adv, desDA3C
```

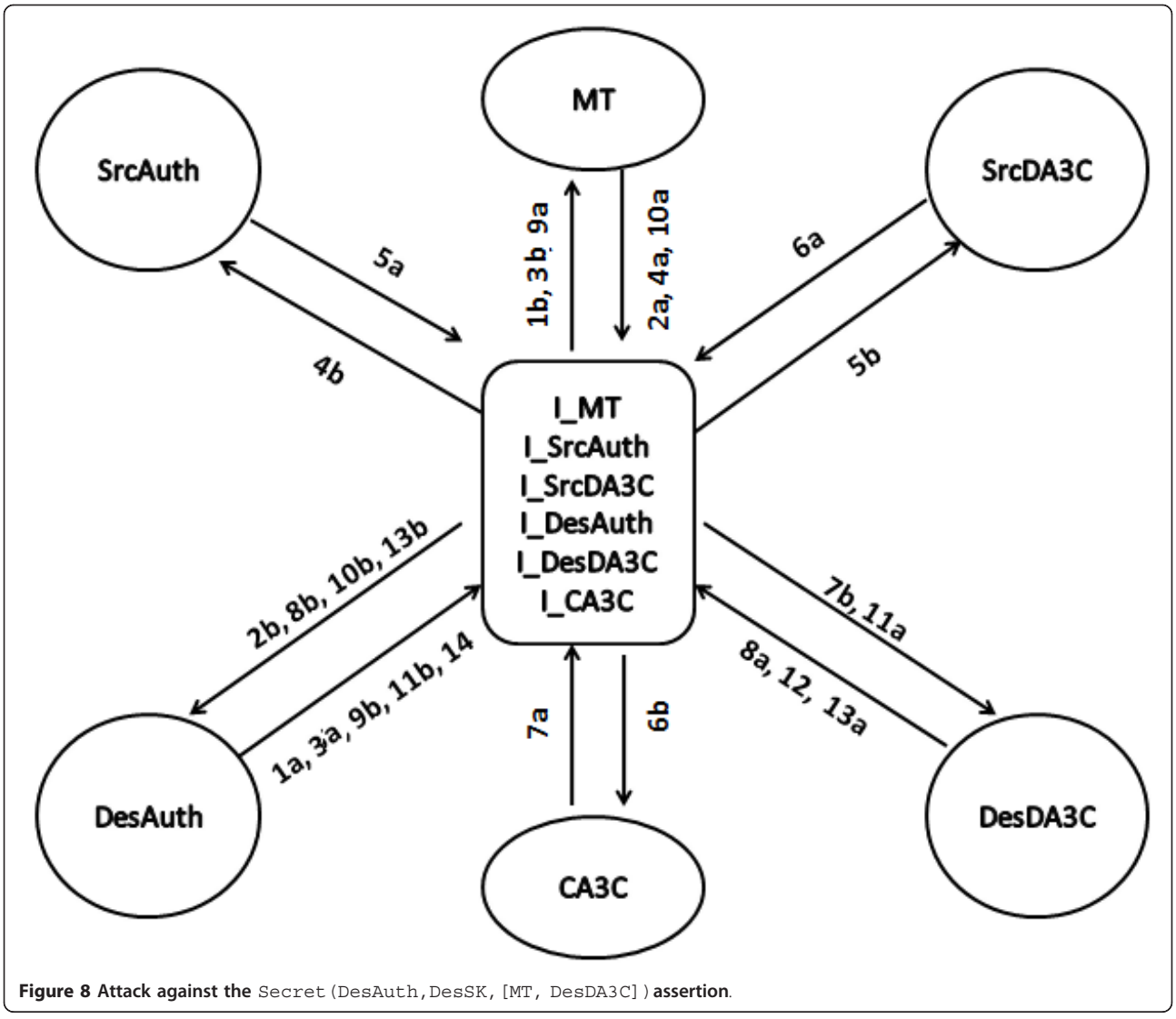

Figure 8 Attack against the `Secret(DesAuth,DesSK,[MT, DesDA3C])` **assertion**.

```
  2a. mt -> I_desAuth : accReq
  2b. I_mt -> desAuth : accReq
  3a. desAuth -> I_mt : authReq
  3b. I_desAuth -> mt : authReq
  4a. mt -> I_srcAuth : {SEQ1, authID, mt,
  initauth}{srcSK}
  4b. I_mt -> srcAuth : {SEQ1, authID, mt,
  initauth}{srcSK}
  5a. srcAuth -> I_srcDA3C : SEQ1, authID,
mt,
  initauth
  5b. I_srcAuth -> srcDA3C : SEQ1, authID,
mt,
  initauth
  6a. srcDA3C -> I_ca3c : {SEQ1, authID, mt,
  initauth}{Srcse1(srcDA3C)}
  6b. I_srcDA3C -> ca3c : {SEQ1, authID, mt,
  initauth}{Srcse1(srcDA3C)}
  7a. ca3c -> I_desDA3C : {DSMS, {SEQ1,
authID,
  mt, initauth}{DSMS}}{Desse1(desDA3C)}
  7b. I_ca3c -> desDA3C : {DSMS, {SEQ1,
authID,
  mt, initauth}{DSMS}}{Desse1(desDA3C)}
  8. desDA3C -> I_desAuth : {SEQ2, SEQ1}
{DesAK}
  9. I_desAuth -> mt : {SEQ2, SEQ1}{DesAK}
  10. mt -> I_desAuth : {SEQ2}{DesAK}
  11. I_desAuth -> desDA3C : {SEQ2}{DesAK}
  12. desDA3C -> I_ca3c : {hoAckm}{Desse1
(desDA3C)}
  13. desDA3C -> I_desAuth : DesSK
```

As shown in Figure 9, the first set of messages (0-7b) are same to the previous secrecy attack. Then, starting

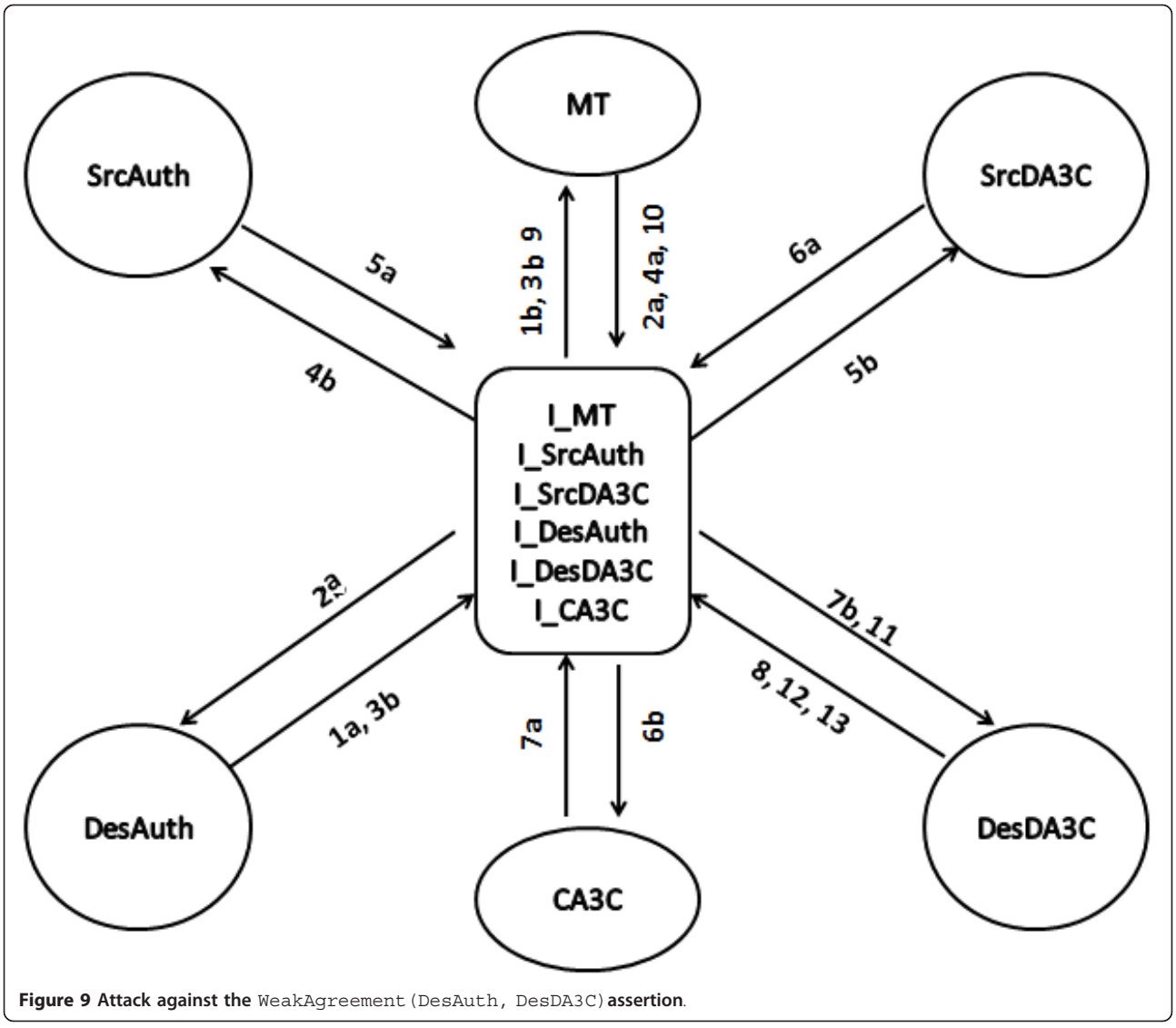

Figure 9 Attack against the `WeakAgreement(DesAuth, DesDA3C)` **assertion**.

from message 8, the intruder impersonates the DesAuth and completes the protocol's run. Therefore, the discovered attack could be interpreted as follows: the DesDA3C believes that it has successfully completed the run with the DesAuth. However, in reality it was with the intruder acting as the DesAuth.

6.4 The second version protocol

As shown in Figure 10, in this version of the protocol, secure channels exist only between the DA3Cs and the Auths. To simulate these channels, secret keys Srcse2 (SrcAuth) and Desse2(DesAuth) are pre-shared between the Auth and the DA3C in the source and destination domains, respectively. After preparing the Casper's input file and asking Casper/FDR to verify the protocol, Casper found the following attack against the `Agreement (DesDA3C, MT, [seq1, DesAK])` assertion.

```
0. -> mt : srcAuth, desAuth, srcDA3C
1a. desAuth -> I_mt : adv, desDA3C
1b. I_desAuth -> mt : adv, desDA3C
2a. mt -> I_desAuth : accReq
2b. I_mt -> desAuth : accReq
3a. desAuth -> I_mt : authReq
3b. I_desAuth -> mt : authReq
4a. mt -> I_srcAuth : {SEQ1, authID, mt,
initauth}{srcSK}
4b. I_mt -> srcAuth : {SEQ1, authID, mt,
initauth}{srcSK}
5a. srcAuth -> I_srcDA3C : {SEQ1, authID,
mt,
initauth}{Srcse2(srcAuth)}
5b. I_srcAuth -> srcDA3C : {SEQ1, authID,
mt,
initauth}{Srcse2(srcAuth)}
```

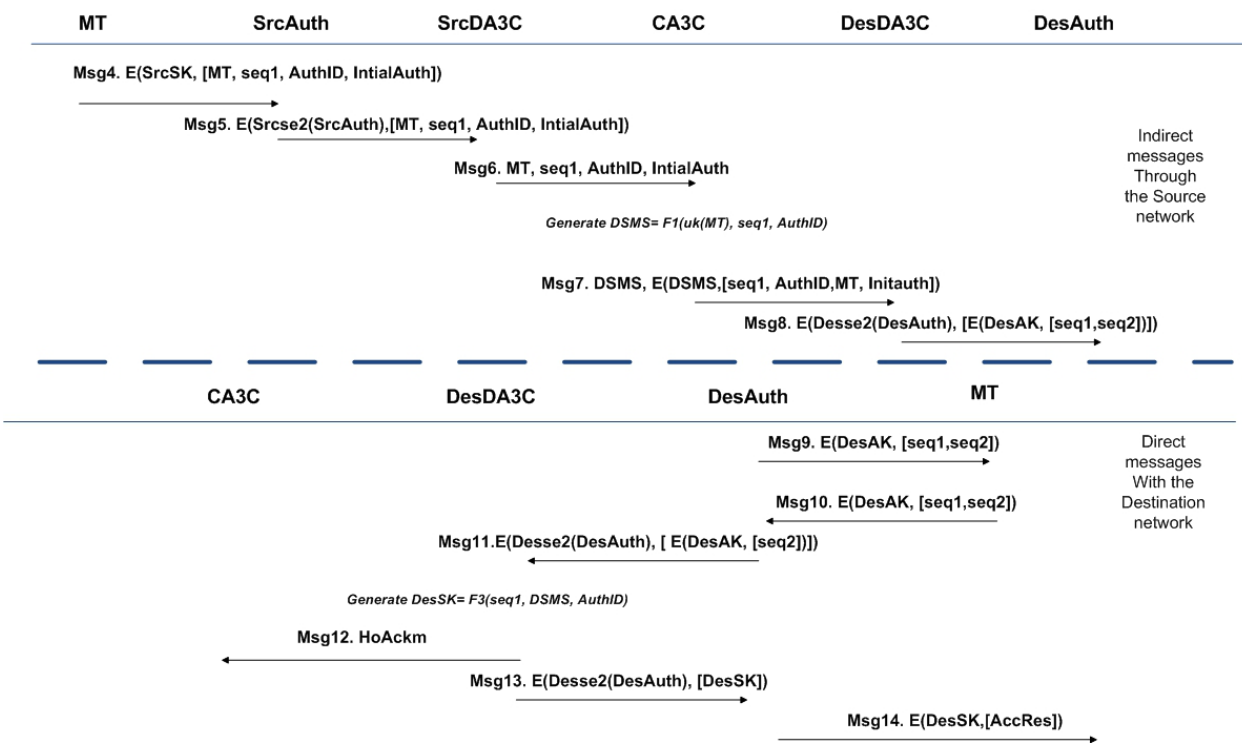

Figure 10 The second version of the proposed AKA protocol.

```
6a. srcDA3C -> I_ca3c : SEQ1, authID, mt,
initauth
6b. I_srcDA3C -> ca3c : SEQ1, authID, mt,
initauth
7a. ca3c -> I_desDA3C : DSMS, {SEQ1,
authID, mt, initauth}{DSMS}
7b. I_ca3c -> desDA3C : DSMS, {SEQ1,
authID,
Mallory, initauth}{DSMS}
8a. desDA3C -> I_desAuth : {{SEQ2, SEQ1}
{DesAK}}
{Desse2(desAuth)}
8b. I_desDA3C -> desAuth : {{SEQ2, SEQ1}
{DesAK}}
{Desse2(desAuth)}
9a. desAuth -> I_mt : {SEQ2, SEQ1}{DesAK}
9b. I_desAuth -> mt : {SEQ2, SEQ1}{DesAK}
10a. mt -> I_desAuth : {SEQ2}{DesAK}
10b. I_mt -> desAuth : {SEQ2}{DesAK}
11a. desAuth -> I_desDA3C : {{SEQ2}
{DesAK}}
{Desse2(desAuth)}
11b. I_desAuth -> desDA3C : {{SEQ2}
{DesAK}}
{Desse2(desAuth)}
12. desDA3C -> I_ca3c : hoAckm
13a. desDA3C -> I_desAuth : {DesSK}
{Desse2
(desAuth)}
13b. I_desDA3C -> desAuth : {DesSK}
{Desse2
(desAuth)}
14a. desAuth -> I_mt : {accRes}{DesSK}
14b. I_desAuth -> mt : {accRes}{DesSK}
```

The above sequence is depicted in Figure 11 and could be explained as follows:

- Similar to the previous attacks, the intruder intercepts the preliminary messages between the MT and the authenticator of the destination networks as in messages (1a-3b). Then, it intercepts and replays messages (4a-7a) in the source domain.
- The intruder fakes message 7a by replacing the MT with its identity (Mallory), and passes it to the DesDA3C as message 7b. The DesDA3C, mistakenly believing that the CA3C has identified Mallory, generates the DesAK and composes message 8a, which will be intercepted by the intruder.
- The intruder intercepts and replays messages (9a, 9b, 10a, 10b, 11a, and 11b) in the destination domain. Once the DesDA3C verifies the contents of message 11b, it mistakenly authenticates the

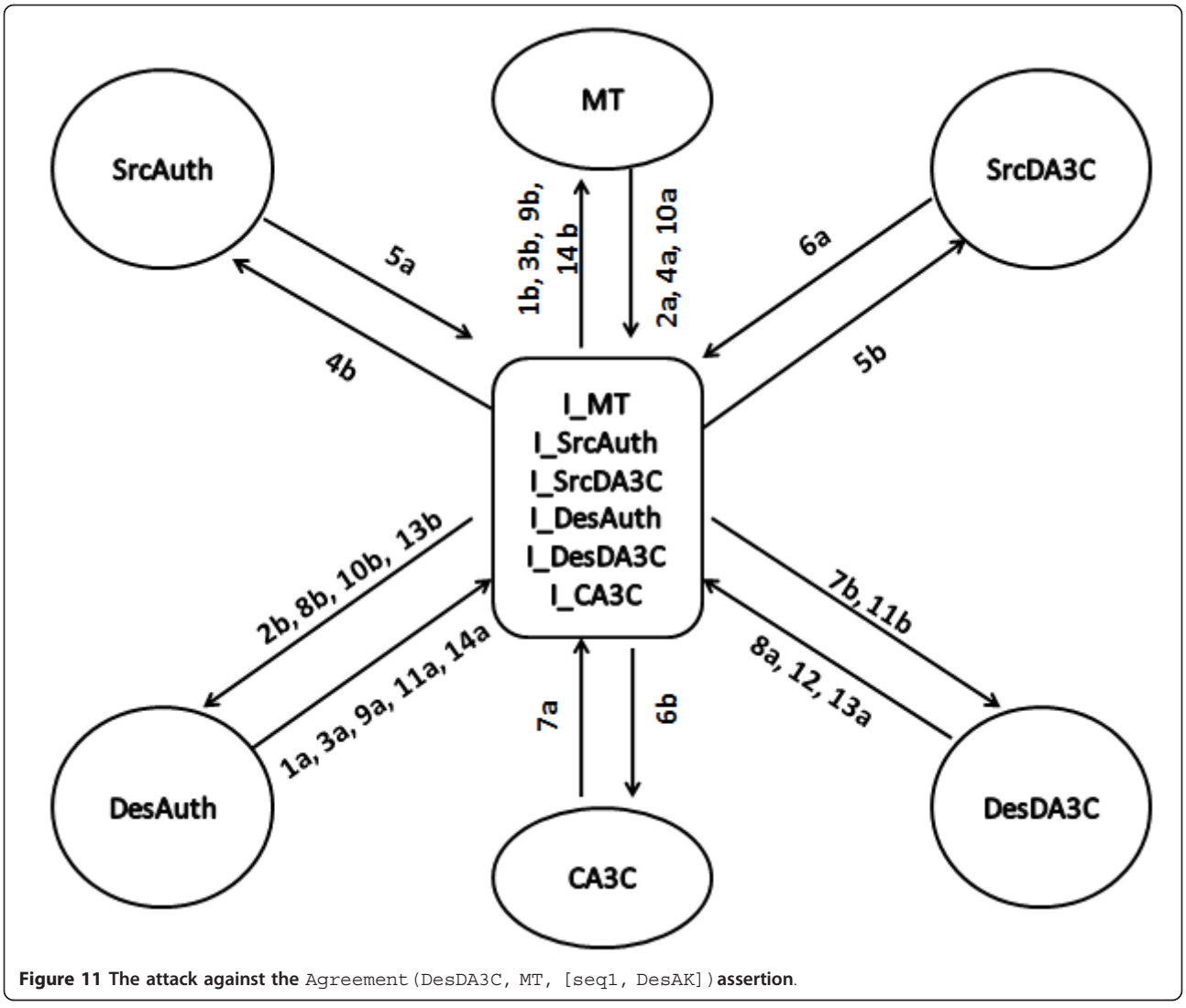

Figure 11 The attack against the `Agreement(DesDA3C, MT, [seq1, DesAK])` **assertion**.

intruder, acknowledges the successful authentication and sends the secret key DesSK as messages 12, 13a, respectively. However, these messages will be blocked by the intruder.

- At this stage, the intruder wants to finish the attack stealthy, so it passes the intercepted DesSK to the DesAuth to generate the AccRes message and finishes the protocol in message 13b, 14a, and 14b.

6.5 The final protocol

The first and second versions of the protocol in the Sections 6.3 and 6.4, highlight the fact that, there is a need to protect all the parts and connections in the core network. Therefore, in this final version of the proposed protocol, secure channels between the Auths and the DA3Cs as well as the between the DA3Cs and the CA3C have been considered, as shown in Figure 12. We simulated this security considerations with Casper and asked FDR to check for attacks. Casper/FDR failed to find attacks against any of the assertions in the #Specifications heading.

This result implies that the assumption in current systems such as 3G and 2G of a physically secure core network could not valid any more. Therefore, in order to provide security in future, heterogeneous environments, there is a need to protect each part and connection in the core network.

6.6 AKA protocol formal verification

The main goal of the proposed protocol is to achieve mutual authentication between the MT and the core network in case of handover, thus authenticating the MT to use the destination peripheral network. To

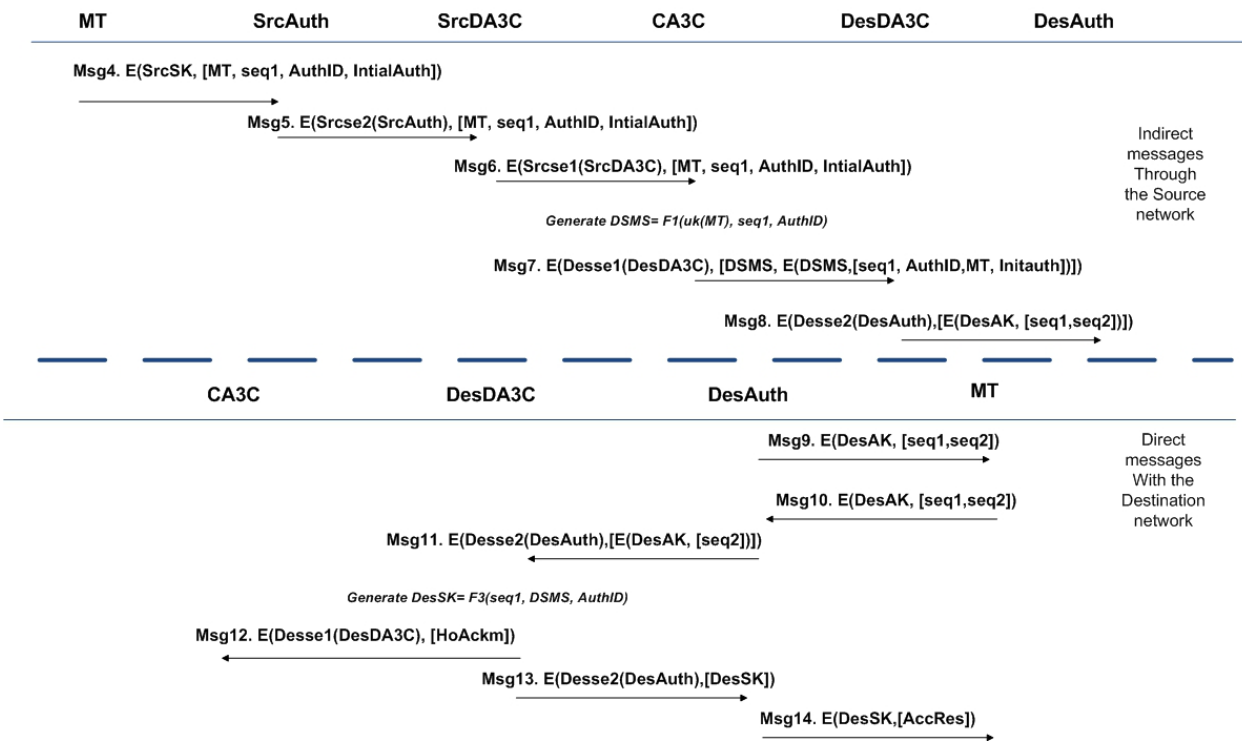

Figure 12 The final version of the proposed AKA protocol.

model the AKA protocol using Casper/FDR tool, we prepared a Casper input file that represents the system. The complete Casper description is in Appendix B.

This section discusses how our formal modeling with Casper allows checking the typical security requirements for AKA security protocols. In this section, we discuss how our formal modeling with Casper allows checking the typical security requirements for AKA security protocols.

- *Mutual entity authentication:* Casper provides no direct specification to model this property. In order to show how our protocol could meet this requirement, we explicitly, and by considering the protocol transactions, could argue that this requirement could be met to a certain extent in our protocol. When making the initial contract, the MT and the CA3C share a unique key uk(MT), which acts as the root in the key hierarchy and is never used for encryption. We assume this key has been derived by running a KDF over identity-related information of the MT and the CA3C, and since it is never exposed and is stored in the MT's SIM card, it is unlikely for an intruder to get that key; thus, possessing this key verifies the identity of the party.
- *Mutual key authentication:* the mutual authentication between the MT and the DesDA3C is based on the secrecy of the freshly derived DesAK. We got Casper to check this using the Secret (MT, DesAK, [DesDA3C]) and Secret (DesDA3C, DesAK, [MT]) assertion checks. Since Casper/FDR found no attacks against the secrecy of the DesAK, this implies that, only other party apart form the intended ones could possess this key.
- *Mutual key confirmation:* Casper verifies this requirement by using the DECRYPTABLE (m, K) which checks if the message (m) is decryptable by the key (K). We performed a similar check after messages 9 and 11 as shown in the Protocol Description heading to verify that the valid AK is possessed by the other party. If any of the checks fails the protocol aborts.
- *Key freshness:* since there is no direct function with Casper to simulate this feature, we included a freshly generated sequence seq1 in the KDF as explained in the key derivation subsection; thus the fact that Casper does not detect any attack on the secrecy of the secret and AKs implies that key freshness is not violated.
- *Unknown-key share resilience:* we check this property using the Aliveness assertions. Additionally, we could address this attack by making a binding between the keys and the identity of the parties. The proposed AKA protocol has achieved this by the

identity of the MT and the CA3C in the derivation of the uk(MT). Also, the authenticator's ID and the domain name are included in the KDFs of the Secret and AKs.

- *Key compromise impersonation resilience:* this property could be modeled by specifying the long-term keys as crackable and then checking the authenticity assertions.

Three more features could be achieved by the proposed protocols these are as follows:

1. *Forward secrecy (FS):* A protocol is said to meet this requirement if the compromise of long-term keys does not compromise past session keys that have been established before the compromise of the long-term key. We got Casper to check this property by specifying the DSMS as crackable (Crackable = Domainspecifickey) in the #Intruder Information and checking the secrecy of the previous SK in the source network (SrcSK) by adding the Secret (SrcAuth,SrcSK,[MT, SrcDA3C]) assertion in the #Specification heading of Appendix B. Since no attack was found against this assertion, we could claim that our protocol meets the FS property.
2. The second feature is whether the compromise of the DSMS will lead to a compromise of the derived keying materials. This feature was checked by specifying DSMS as crackable and checking the secrecy of the DesAK and DesSK in assertions Secret(MT, DesAK, [DesDA3C]) and Secret(DesAuth, DesSK, [MT, DesDA3C], respectively. We had run this check and could confirm that, no attacks were found.
3. The third feature is whether the compromise of one of the SKs will lead to the compromise of the other SKs. We tested this by adding the SrcSK key to the Intruder Knowledge and checked the secrecy of the new secret key (DesSK). No attacks were found and thus we could confirm that, this protocol meets this requirement.

6.7 Protocol analysis and security consideration

Table 4 shows a summary of the results, it compares between the mobile ethernet's AKA protocol, the first, second and final versions of our proposed solution.

7 Further work

The research in this article aims at providing a platform-independent AKA protocol that could be implemented by a wide variety of network operators. However, since some network providers deploy the EAP as a platform on top of which different types of security protocols could run, the authors want to consider integrating the proposed AKA protocol as an EAP method. Thus, operators will have the choice to use the proposed protocol as an EAP method or as a pure AKA protocol.

Having had the AKA protocol verified by Casper, the next step will be implementing the protocol by using compilers like COSP-J [29] which is a compiler that takes a description of a security protocol in a simple, abstract language, and produces a Java implementation of the it. In addition, implementing the protocols on smart phones gives us a chance to measure the performance of the proposed protocol in real test-bed as well as to discover any implementation-based attacks.

Furthermore, study has already started within our group to propose a business model which will define charging and accounting models to charge mobile devices in heterogeneous networks.

8 Conclusion

This article discussed several research efforts, which have been trying to address the issue of authenticating the mobile nodes when they perform vertical handover in heterogeneous environment. The discussion showed that most of the solutions had realized the threats resulting from the open nature of future networks and as a result different approaches were proposed. Some solutions tried to conceal the divergence of the core network either by considering a specific technology as a backbone of the core network, or by deploying a common framework on top of which security protocols could be installed and run. The mobile ethernet group proposed a new AKA, which considers an open network architecture. Analyzing and verifying the mobile ethernet's AKA protocol using Casper/FDR shows that the protocol is vulnerable to an authentication attack. Also, the protocol failed to meet some desired security properties, which could be ascribed to the lack of security in the core network. Therefore, a new AKA protocol was introduced in this article, the article described the refinement stages of the protocol along with the discovered attacks. The final version of the proposed protocol was proven to be secure and to fulfill the desired security properties.

Appendix A: Code for formal analysis of the handover AKA protocol for mobile ethernet

```
# Free Variables M: MobileTerminal
  EP : AccessRouterAuthenticator
  AS : DomainA3CServer
  AuthID : Identity
  Initauth : Flags
  R1 : initialSeq
  R2 : Sequence
  HOAID1: OldToken
```

Table 4 Comparison

The security property	The AKA of mobile ethernet	Initial version	Second version	Refined proposal
Mutual entity authentication	No	Yes	Yes	Yes
Mutual key authentication/keys' secrecy	Yes	No	Yes	Yes
Mutual key confirmation	Yes	No	Yes	Yes
Key freshness	Yes	Yes	Yes	Yes
Unknown-key share **resilience**	No	No	No	Yes
Key compromise impersonation resilience	Yes	Yes	Yes	Yes
Defining key scope	No	Yes	Yes	Yes

```
HOAID2: NewToken
AK : AuthenticationKeys
SK : SecretKeys
MS: Domainspecifickey
F: AuthenticationKeys × initialSeq × Sequence ->
NewToken
h : HashFunction
AccReq, AccRes,AuthReq, Adv: Messages
HoAckm : AcknowledgementMessage
InverseKeys = (AK, AK), (SK, SK), (MS, MS), (F,F)
# Processes
INITIATOR(M, EP, R1,AuthID,Initauth, AccReq, AuthReq, MS, AK, SK, HOAID1)
Authenticator(EP,AS, AuthReq, Adv, AccRes)
DomainSERVER(AS,M, R2, HoAckm, MS, AK, SK, HOAID1)
# Protocol Description
0. -> M : EP, AS
1. M -> EP: AccReq
2. EP -> M : AuthReq
3. M -> EP : {M,R1, HOAID1}{SK}%w
4. EP -> AS : w%{M,R1,HOAID1}{SK}, h(w%{M,R1,
HOAID1}{SK})
5. AS -> EP: {R2,{R1}{AK}%z}{SK}%v
6. EP -> M : v%{R2,{R1}{AK}%z}{SK}
[decryptable(z, AK)andnth(decrypt(z, AK), 1) == R1]
# Specification
Secret (M,AK,[AS])
Secret (AS,AK,[M])
Secret (M,SK,[AS, EP])
Agreement (AS, M, [AK, R1])
WeakAgreement (M, EP)
WeakAgreement (EP,M)
Aliveness (EP, M)
Aliveness (M, EP)
# Actual Variables
m, Eve: MobileTerminal
ep : AccessRouterAuthenticator
as : DomainA3CServer
Authid : Identity
InitAuth : Flags
hoaid1: OldToken
hoaid2: NewToken
r1 : initialSeq
r2 : Sequence
ak : AuthenticationKeys
sk : SecretKeys
ms: Domainspecifickey
accReq, accRes,authReq, adv: Messages
hoackm : AcknowledgementMessage
InverseKeys = (ms, ms), (ak, ak), (sk, sk)
# Functions
symbolic F
# System
INITIATOR (m,ep, r1,Authid,InitAuth, accReq, authReq, ms, ak,sk,hoaid1)
Authenticator (ep, as, authReq,adv, accRes)
DomainSERVER (as,m, r2,hoackm, ms,ak,sk, hoaid1)
# Intruder Information
Intruder = Mallory
IntruderKnowledge = m, as, Mallory, Authid, ep
Crackable = Domainspecifickey
```

Appendix B: Code for formal analysis of the proposal handover AKA protocol

```
# Free Variables
MT: Agent
SrcAuth : SrcAccessRouterAuthenticator
DesAuth : DesAccessRouterAuthenticator
SrcDA3C : SrcDomainA3CServer
DesDA3C : DesDomainA3CServer
CA3C : CentralA3CServer
AuthID : Identity
Initauth : Flags
seq1 : initialSeq
seq2 : Sequence
Srcse1 : SrcDomainA3CServer-> PresharedKeys
```

Desse1 : DesDomainA3CServer-> PresharedKeys
Srcse2 : SrcAccessRouterAuthenticator-> PresharedKeys
Desse2 : DesAccessRouterAuthenticator-> PresharedKeys
uk : Agent-> PresharedKeys
SrcAK, DesAK : AuthenticationKeys
SrcSK, DesSK : SecretKeys
DSMS: Domainspecifickey
AccReq, AccRes,AuthReq, Adv: Messages
HoAckm : AcknowledgementMessage
F1: PresharedKeys × initialSeq × Identity -> Domainspecifickey
F2: initialSeq × Domainspecifickey -> AuthenticationKeys
F3: initialSeq × Domainspecifickey × Identity -> SecretKeys
InverseKeys = (SrcAK, SrcAK), (uk, uk), (SrcSK, SrcSK),
(DSMS, DSMS), (Srcse1,Srcse1),(Srcse2, Srcse2),
(Desse1,Desse1),(Desse2, Desse2), (DesAK,DesAK),(DesSK, DesSK), (F1, F1), (F2,F2),(F3,F3)

 # Processes

INITIATOR(MT,seq1,AuthID,Initauth, SrcAK, SrcSK, AccReq) knows uk(MT)
SrcAuthenticator(SrcAuth,MT,SrcDA3C, SrcSK, AuthReq)
knows Srcse2(SrcAuth)
DesAuthenticator(DesAuth,MT, DesDA3C, AuthReq, Adv,
AccRes) knows Desse2(DesAuth)
SrcAAASERVER(SrcDA3C,CA3C, SrcAuth, SrcAK, SrcSK) knows Srcse1(SrcDA3C), Srcse2(SrcAuth)
DesAAASERVER(DesDA3C,CA3C,DesAuth, seq2,HoAckm) knows Desse1(DesDA3C), Desse2(DesAuth)
CentralSERVER(CA3C, SrcDA3C, DesDA3C) knows
Srcse1(SrcDA3C), Desse1(DesDA3C), uk (MT)

 # Protocol Description

0. -> MT : SrcAuth, DesAuth, SrcDA3C
1. DesAuth -> MT :Adv, DesDA3C
< *DSMS* := *F*1(*uk*(*MT*), *seq*1, *AuthID*) >
2. MT -> DesAuth: AccReq
3. DesAuth -> MT : AuthReq
< *DesAK* := *F*2(*seq*1, *DSMS*) >
4. MT -> SrcAuth : {seq1,AuthID, MT, Initauth}{SrcSK}
5. SrcAuth -> SrcDA3C : {seq1,AuthID, MT, Initauth}{
Srcse2(SrcAuth)}
6. SrcDA3C -> CA3C : {seq1,AuthID, MT, Initauth}{
Srcse1(SrcDA3C)}
< *DSMS* := *F*1(*uk*(*MT*), *seq*1, *AuthID*) >
7. CA3C -> DesDA3C : {DSMS, {seq1,AuthID, MT, Initauth}
{DSMS}}{Desse1(DesDA3C)}
< *DesAK* := *F*2(*seq*1, *DSMS*) >
8. DesDA3C -> DesAuth: {({seq2, seq1}{DesAK}%z)%x}
{Desse2(DesAuth)}
9. DesAuth -> MT : x%({seq2, seq1}{DesAK}%z)
[*decryptable*(*z*, *DesAK*)*andnth*(*decrypt*(*z*, *DesAK*), 2) == *seq*1]
< *DesSK* := *F*3(*seq*1, *DSMS*, *AuthID*);
*seq*2 := *nth*(*decrypt*(*z*, *DesAK*), 1) >
10. MT -> DesAuth : ({seq2}{DesAK}%y)%q
11. DesAuth -> DesDA3C: {(q% seq2}{DesAK})%y}
{Desse2(DesAuth)}
[*decryptable*(*y*, *DesAK*)*andnth*(*decrypt*(*y*, *DesAK*), 1) == *seq*2]
< *DesSK* := *F*3(*seq*1, *DSMS*, *AuthID*) >
12. DesDA3C -> CA3C : {HoAckm}{Desse1(DesDA3C)}
13. DesDA3C -> DesAuth :{DesSK}{Desse2(DesAuth)}
14. DesAuth -> MT : {AccRes}{DesSK}

 # Specification

Secret(MT,DesAK,[DesDA3C])
Secret(DesAuth,DesSK,[MT, DesDA3C])
Secret(SrcAuth,SrcSK,[MT, SrcDA3C])
Agreement(MT, DesDA3C, [seq2])
Agreement(DesDA3C, MT, [seq1, DesAK])
WeakAgreement(MT, DesAuth)
WeakAgreement(DesAuth, MT)
WeakAgreement(DesAuth, DesDA3C)
WeakAgreement(DesDA3C, DesAuth)
Aliveness(MT, DesAuth)
Aliveness(DesAuth, MT)

 # Actual Variables

mt, Mallory: Agent
srcAuth : SrcAccessRouterAuthenticator
desAuth : DesAccessRouterAuthenticator
srcDA3C : SrcDomainA3CServer
desDA3C : DesDomainA3CServer
ca3c : CentralA3CServer
authID : Identity
initauth : Flags
SEQ1 : initialSeq

```
  SEQ2 : Sequence
  srcAK, desAK : AuthenticationKeys
  srcSK, desSK : SecretKeys
  dsms: Domainspecifickey
  accReq : AccessReqmessages
  accRes: AccessResmessages
  authReq: Authmessage
  adv:AdvMessages
  hoAckm : AcknowledgementMessage
  InverseKeys = (dsms, dsms), (srcAK,
srcAK), (srcSK, srcSK), (desAK, desAK),
(desSK, desSK)
  # Functions
  symbolic Srcse1,Srcse2,Desse1,Desse2,
uk, F1, F2, F3
  # System
  INITIATOR(mt,SEQ1,authID,initauth,
srcAK, srcSK,accReq)
  SrcAuthenticator(srcAuth, mt,srcDA3C,
srcSK, authReq)
  DesAuthenticator(desAuth, mt, desDA3C,
authReq,adv, accRes)
  SrcAAASERVER(srcDA3C,ca3c,srcAuth,
srcAK, srcSK)
  DesAAASERVER(desDA3C,ca3c,desAuth,
SEQ2,hoAckm)
  CentralSERVER(ca3c, srcDA3C, desDA3C)
  # Intruder Information
  Intruder = Mallory
  IntruderKnowledge = {mt, srcDA3C,des-
DA3C, Eve, ca3c, authID, srcAuth,desAuth,
uk(Eve)}
  Crackable = PresharedKeys
  Crackable = Domainspecifickey
```

Author details
[1]School of Engineering and Information Systems (EIS), Middlesex University, London NW4 4BT, UK [2]School of Electronic, Electrical and Systems Engineering, Loughborough University, Loughborough, UK

Competing interests
The authors declare that they have no competing interests.

Received: 14 September 2011 Accepted: 22 February 2012
Published: 22 February 2012

References

1. S Jochen, *Mobile Communications* (Pearson Education Limited, UK, 2003)
2. Long Term Evolution Protocol Overview, Freescale Semiconductor (2008), http://www.freescale.com/files/wireless_comm/doc/white_paper/LTEPTCLOVWWP.pdf. Accessed 19 August 2011
3. M Aiash, G Mapp, A Lasebae, A QoS framework for Heterogeneous Networking, in *ICWN2011*, London UK, 1765–1769 (2011)
4. Internet Engineering Task Force, Handover keying working group (hokey wg) http://www.ietf.org/html.charters/hokey-charter.html. Accessed 19 August 2011
5. 3rd Generation Partnership Project (3GPP), http://www.3gpp.org/. Accessed 19 August 2011
6. Institute of Electrical and Electronics Engineers, IEEE 802.21/D8.0, Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services. (2007)
7. K Masahiro, Y Mariko, O Ryoji, K Shinsaku, T Tanaka: Secure service and network framework for mobile ethernet. Wirel Personal Commun. **29**, 161–190 (2004)
8. 3rd Generation Partnership Project, 3GPP Technical Specifications.: 3G Security; WLAN interworking security (Release 7). (2006)
9. AS Ali, Authentication and key management in heterogeneous wireless networks, PhD Thesis, Electrical and Computer Engineering (The University of British Columbia, 2010)
10. B Aboba, L Blunk, J Vollbrecht, J Carlson, H Levkowetz, *Extensible Authentication Protocol (EAP)* RFC 3748 (June 2004)
11. I KENTARO, K Masahiro, Mobile Ethernet Technologies: 5-1: Scalable Mobile Ethernet and Fast Vertical Handover, in *Review of the National Institute of Information and Communications Technology*. **52**(4), 65–74 (2007)
12. P Ryan, S Schneider, M Goldsmith, G Lowe, AW Roscoe, *The modelling and analysis of security protocols*, (PEARSON Ltd, 2010)
13. Formal Systems (Europe) Ltd., *Failures-Divergence Refinement. FDR2 User Manual*, http://www.fsel.com/documentation/fdr2/fdr2manual.pdf. Accessed 19 August 2011
14. S Xu, C Tser Huang, MM Matthews, Modeling and analysis of IEEE 802.16 PKM Protocols using CasperFDR, in *Wireless Communication Systems, ISWCS'08*, Reykjavik, Iceland, 653–657 (2008)
15. KV Krishnam Raju, V Valli Kumari, Formal verification of IEEE802.11i WPA-GPG authentication protocol. Commun Comput Inf Sci. **147**, 267–272 (2011). doi:10.1007/978-3-642-20573-6_44
16. KV Krishnam Raju, V Valli Kumari, N Sandeep Varma, KVSVN Raju, Formal verification of IEEE802.16m PKMv3 protocol using CasperFDR. Commun Comput Inf Sci. **101**, 590–595 (2010). doi:10.1007/978-3-642-15766-0_101
17. G Lowe, P Broadfoot, C Dilloway, M Hui, Casper, a compiler for the Analysis of security protocol, http://www.comlab.ox.ac.uk/gavin.lowe/Security/Casper/. Accessed 19 August 2011
18. International Telecommunication Union (ITU-T), Global Information Infrastructure, Internet Protocol Aspects And Next Generation Networks, Y.140.1. (2004)
19. S Sargento, V Jesus, F Sousa, F Mitrano, T Strauf, C Schmoll, J Gozdecki, G Lemos, M Almeida, D Corujo, Context-Aware End-to-End QoS Architecture in Multi-technology, Multi-interface Environments, in *16 th Mobile and Wireless Communications Summit*, Budapest 1–6 (2007)
20. IEEE P802.21/D14.0 Media Independent Handover Services (Sept. 2008)
21. U Horn, C Prehofer, H Karl, Ambient networks: an architecture for communication networks beyond 3G. IEEE Wirel Commun, 14–22 (2004)
22. Y-Comm Research, http://www.mdx.ac.uk/research/areas/software/ycomm_research.aspx. Accessed 19 August 2011
23. V Narayanan, L Dondeti, *EAP Extensions for EAP Re-authentication Protocol (ERP)* RFC 5296 (August 2008)
24. IEEE Standard for local and metropolitan area networks, Air Interface for Fixed Broadband Wireless Access Systems, Part 16, Amendment 2 and Corrigendum 1. IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor1-2005
25. WiMAX Forum Network Architecture Stage 3.: Detailed Protocols and Procedures. WiMAX Forum, Rel. 1, ver. 1.2 (January 2008)
26. A Menezes, P van Oorschot, S Vanstone, *Handbook of Applied Cryptography*, (CRC Press, Boca Raton, FL, USA, 1996)
27. P Chandra, *Bulletproof Wireless Security: GSM, UMTS, 802.11 and Ad Hoc Security* (Newnes, Oxford, 2005), pp. 129–158
28. S Kent, R Atkinson, *Security Architecture for the Internet Protocol* RFC 2401 (1998)
29. D Xavier, COSP-J: A compiler for security protocols, Master Thesis, Oxford University Computing Laboratory, http://www.cs.ox.ac.uk/gavin.lowe/Security/Casper/COSPJ/secu.pdf

doi:10.1186/1687-1499-2012-57
Cite this article as: Aiash *et al.*: **A formally verified AKA protocol for vertical handover in heterogeneous environments using Casper/FDR.** *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:57.